Sensor Model Language (SensorML) and Sensor Web Enablement (SWE)

Net Ready Sensors Workshop
ORNL - July 2006

Mike Botts
mike.botts@uah.edu
Principal Research Scientist
University of Alabama in Huntsville

OGC®
Open Geospatial Consortium, Inc.

UAH The University of Alabama in Huntsville

SensorML

What is SensorML?

UAH The University of Alabama in Huntsville

- **Models and XML encodings for describing processes**
  - Detectors, actuators, etc. are modeled as processes
  - Can be used to describe “left-side” processes (“how were these observations obtained?”) and “right-side” processes (“what can I derive from these observations?”)
- **Vision:**

  - **Discovery of transducers and processes / plug-n-play sensors** – SensorML is the means by which sensors and processes make themselves and their capabilities known
  - **Observation lineage** – SensorML provides sensing and processing history of observations; supports quality knowledge of observations
  - **On-demand processing** – SensorML supports on-demand derivation of higher-level information (e.g. geolocation or products) without a priori knowledge of the sensor system
  - **Intelligent, autonomous sensor network** – SensorML enables the development of taskable, adaptable sensor networks

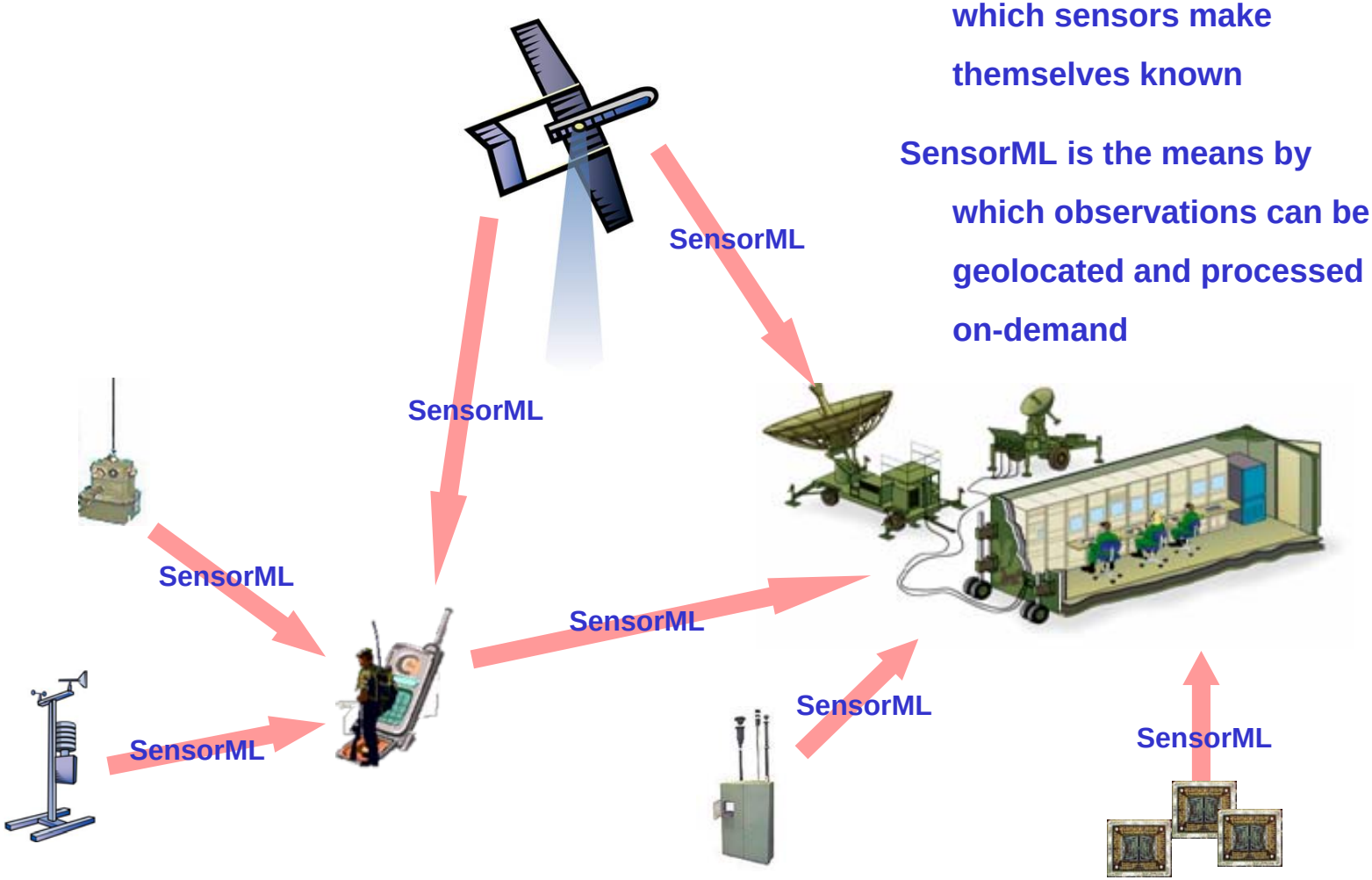
  - **Network and software friendly** – SensorML is ideal for distributing processing demands throughout the network/command hierarchy
  - **Extensibility** – SensorML provides easy means for meeting various community needs

Real-Time Management of Sensor Web Assets

UAH The University of Alabama in Huntsville

SensorML is the means by which sensors make themselves known

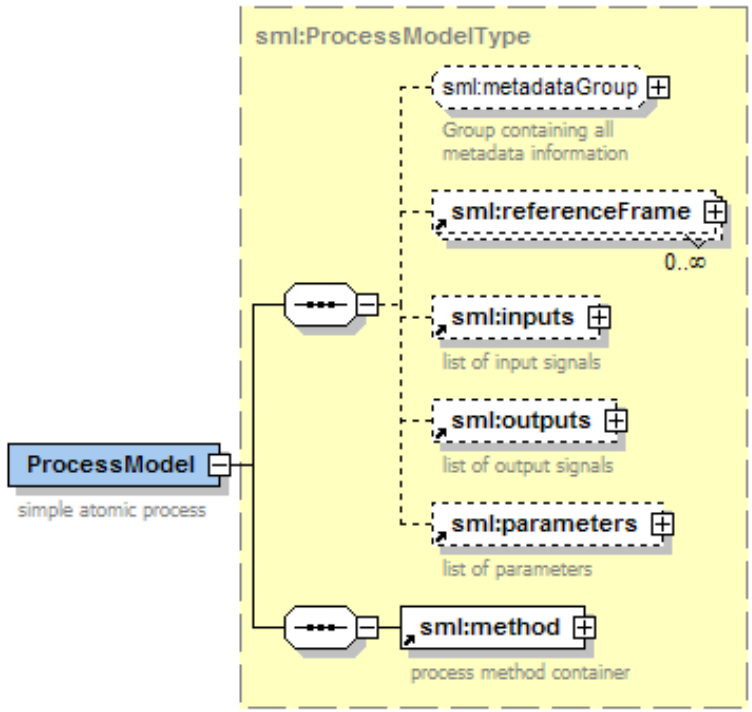
SensorML is the means by which observations can be geolocated and processed on-demand



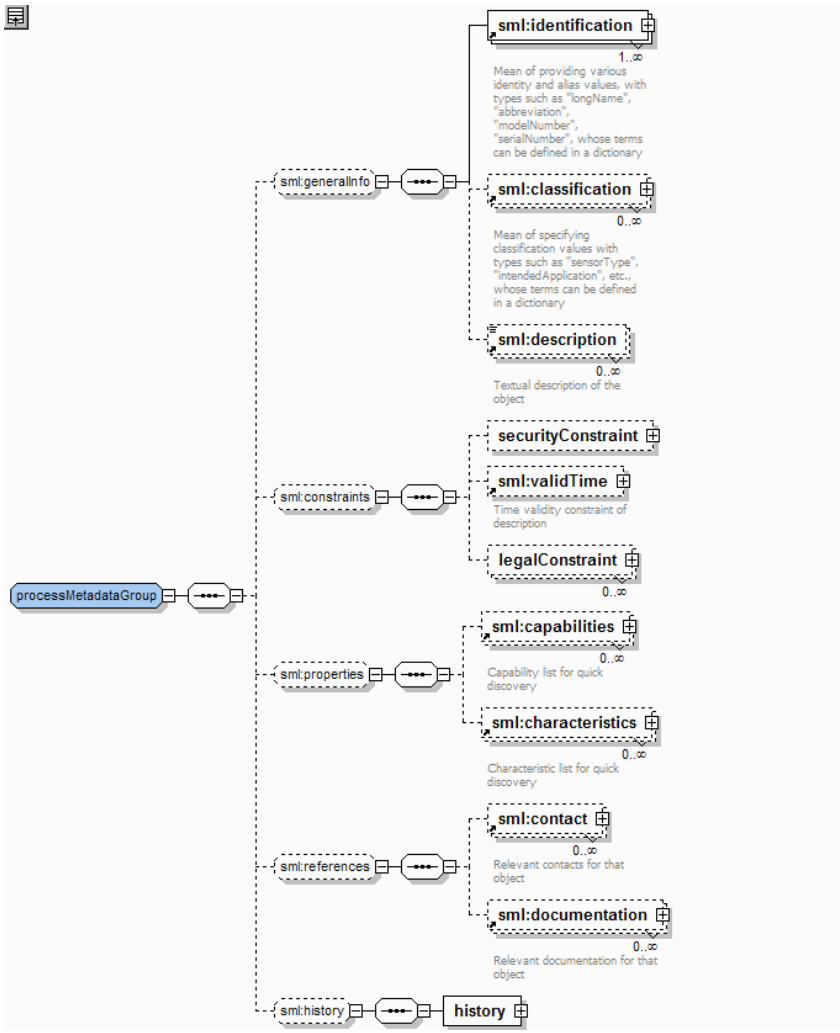
SensorML Process Model (Normative)

In SensorML, everything is modeled as a Process

- **ProcessModel / Component**
  - defines atomic process modules (detector being one)
  - has five sections
    - metadata
    - inputs, outputs, parameters
    - method
  - Inputs, outputs, and parameters defined using SWE Common data definitions



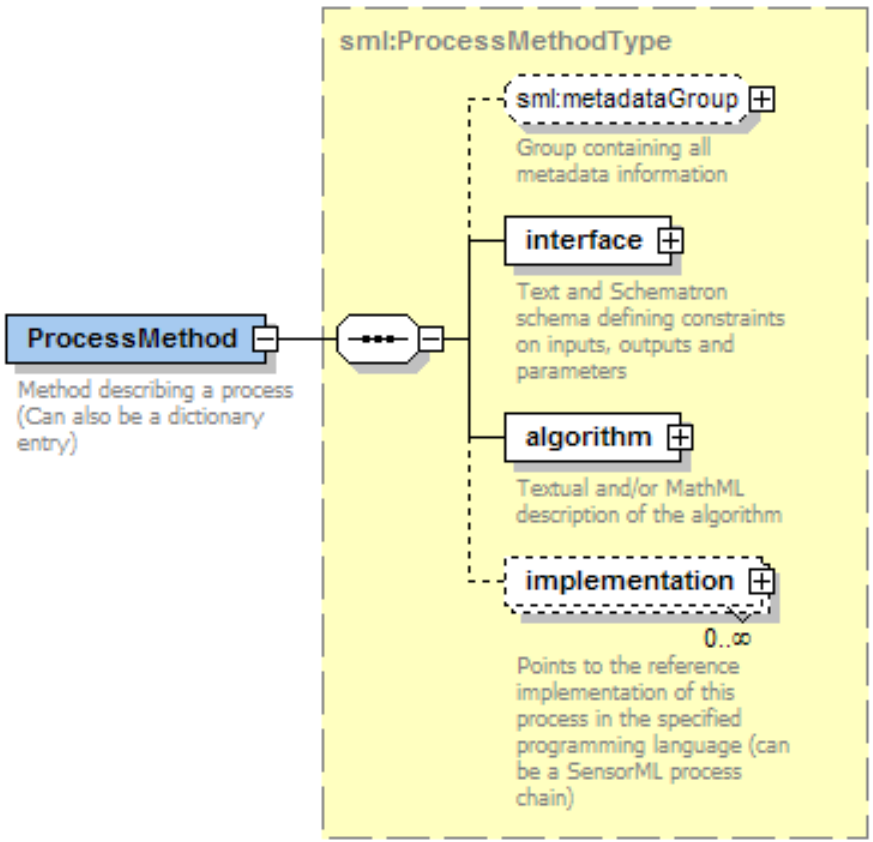
Process/Sensor/System Metadata (Normative)



Metadata is primarily for discovery and assistance, and not typically used within process execution

- **Includes**
  - Identification, classification, description
  - Security, legal, and time constraints
  - Capabilities and characteristics
  - Contacts and documentation
  - History

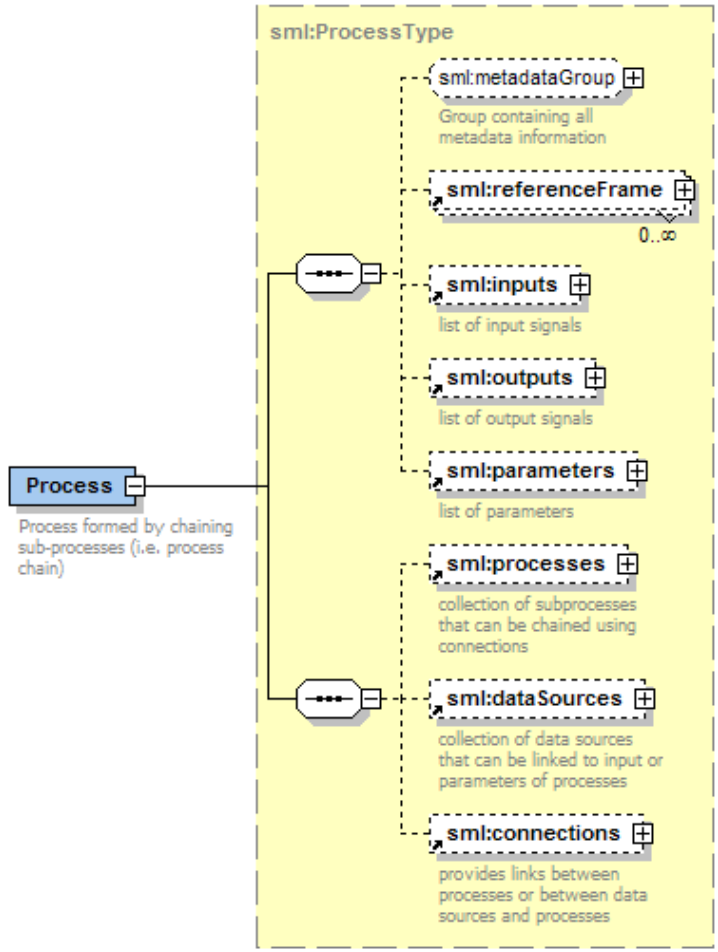
Process Method



- **Metadata group** has same properties as ProcessModel
- **Interface** supports schematron-based validation of XML instance
- **Algorithm** allows textual or MathML description of process
- **Implementation** points to various implementation
  - Java, C, MathML
  - WPS implementation
  - CSM API implementation

SensorML Process Chain (Normative)

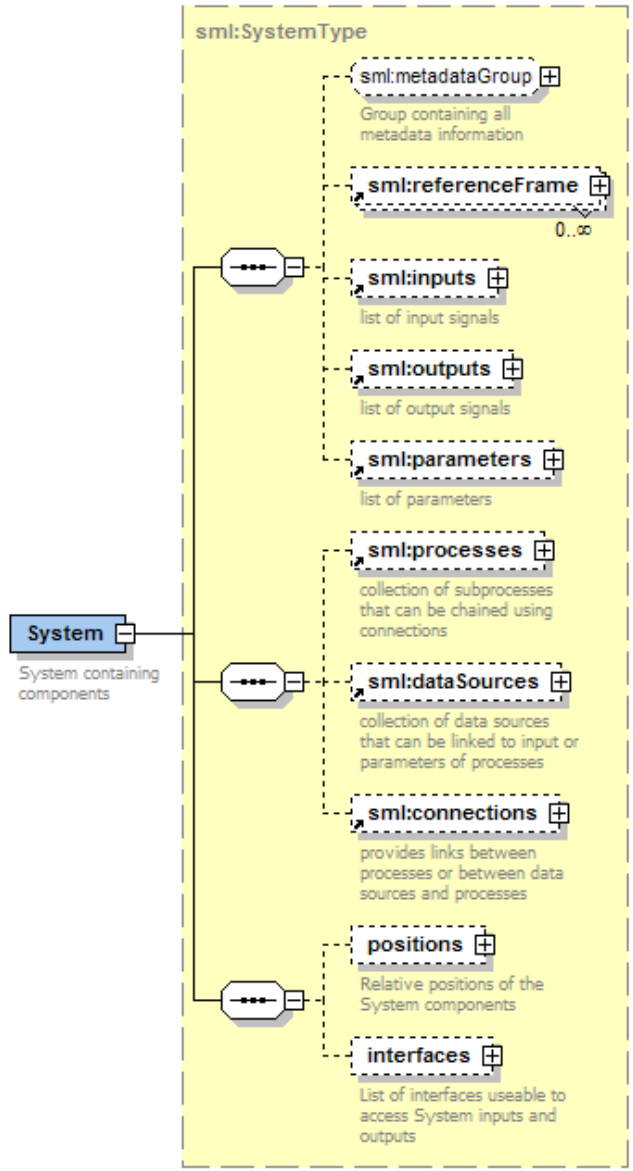
- **ProcessChain**
  - defines a process chain
  - includes:
    - metadata
    - inputs, outputs, and parameters
    - processes (ProcessModel, Process)
    - data sources
    - connections between processes and between processes and data
- **System**
  - defines a collection of related processes along with positional information



SensorML System (Normative)

System

- A system relates a collections of processes to the real world
- Provides relative position of components by defining the relationship between their coordinate frames
- Provides a description of the System interfaces for command and output



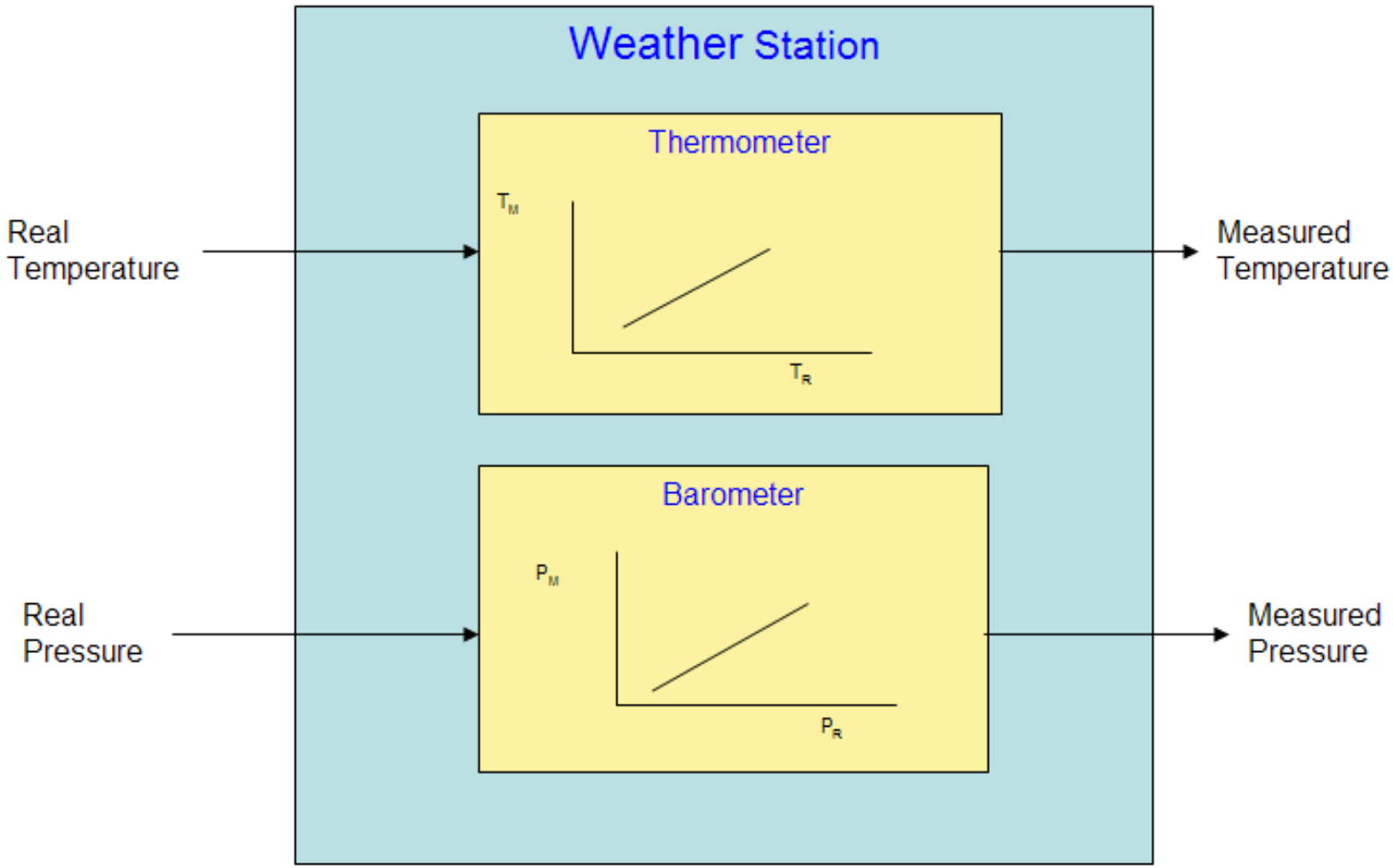
How SensorML can be used -1-

- **“Left-side” versus “Right-side” use**
  - **“Left-Side”**
    - Describes how existing data was obtained
    - Traditional approach
    - Relies on software that has knowledge of sensor system and how to apply particular parameters
    - Useful for defining lineage of observations
    - Useful for simulations of sensor system
  - **“Right-Side”**
    - Describes how to take existing data and derive on-demand higher level information
    - Our vision on how we support discovery and processing of previously unknown observations without a priori knowledge of the sensor system
    - Can utilize process models and parameters from left-side processes

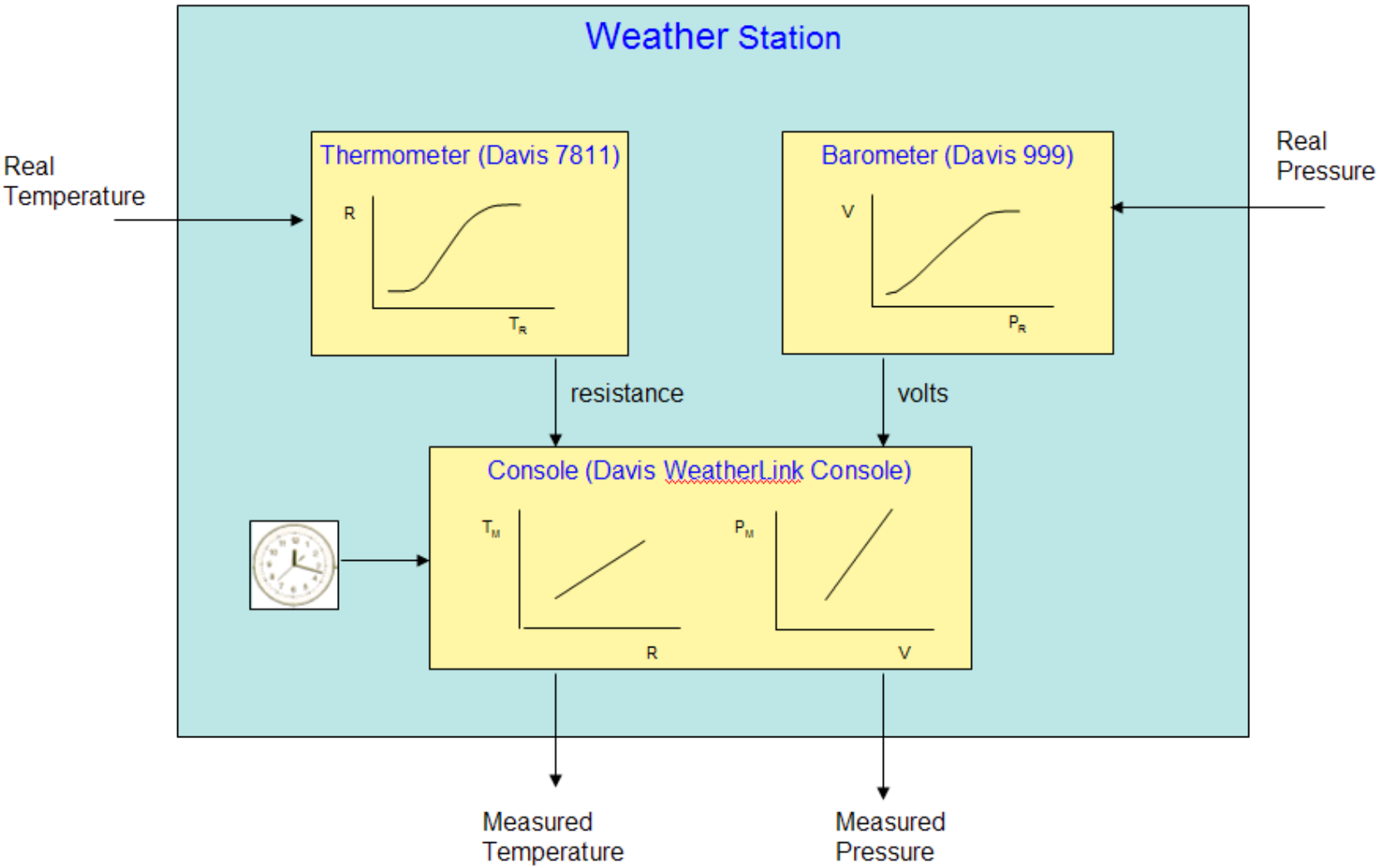
How SensorML can be used -2-

- **Time instance of process versus process factory**
  - **Time instance of process**
    - Typically might accompany a particular instance of an observation and be valid for that time
    - All parameter values for the process are defined inline
    - SPOT DIMAP and NITF approach
  - **Process Factory**
    - Process is defined but variable parameter values may not be specified
    - One treats process like factory assembly line; supply appropriate parameter values and then pump through the input values to obtain appropriate output values

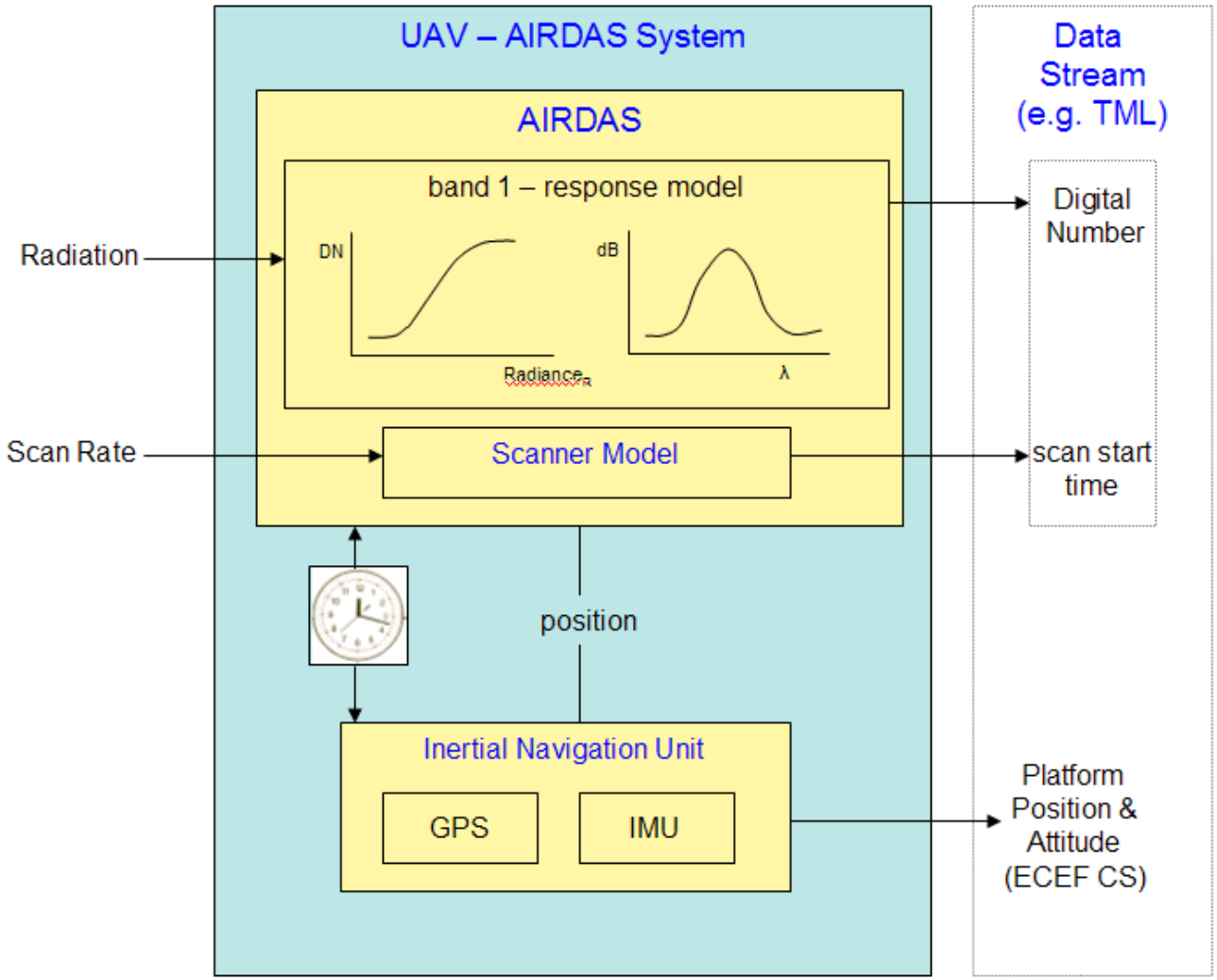
System Example – Simple Weather Station



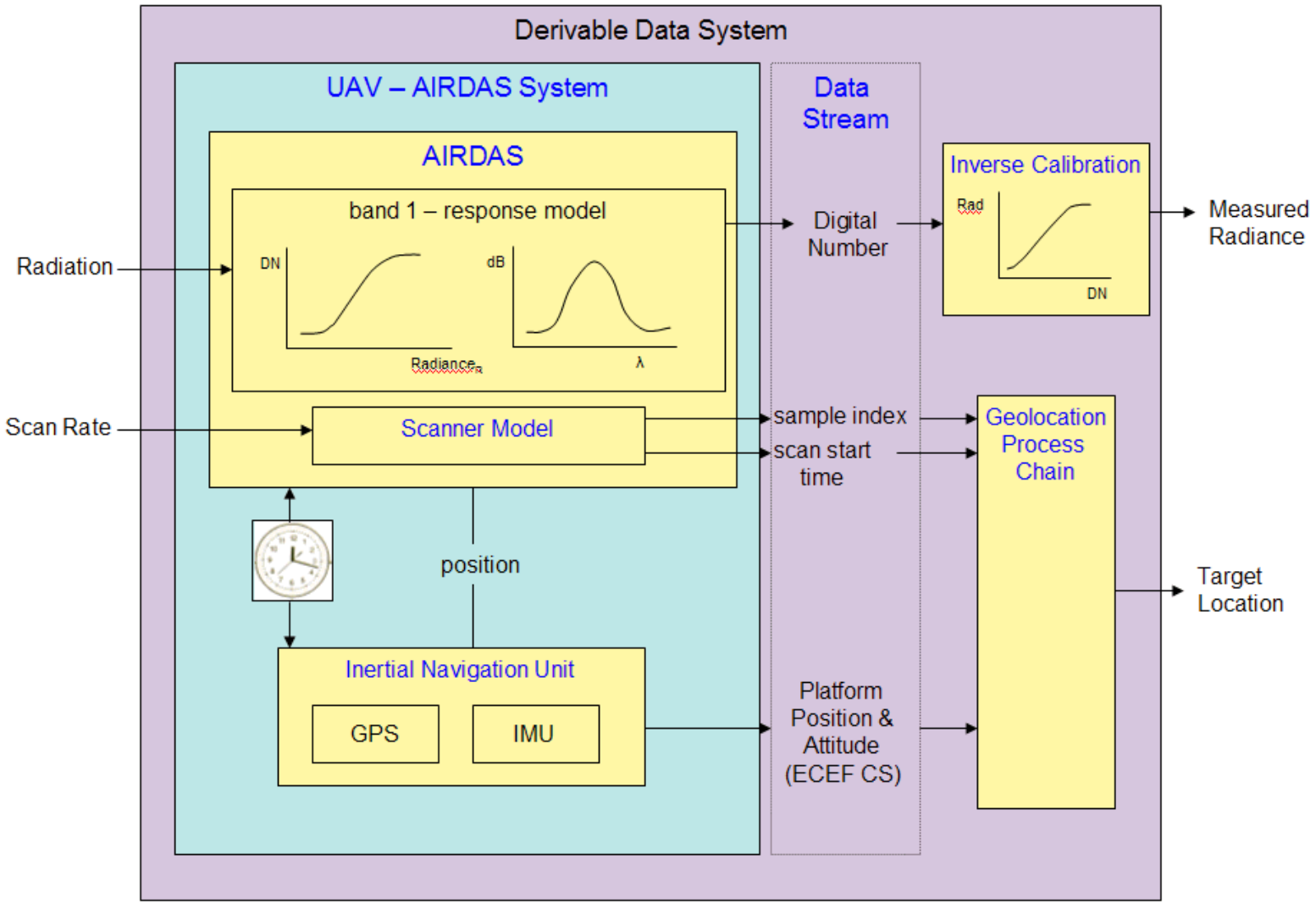
System Example – More explicit weather station



System Example – Basic UAV-AIRDAS system



System Example – AIRDAS with derivable data



Example Process Models (Informative)

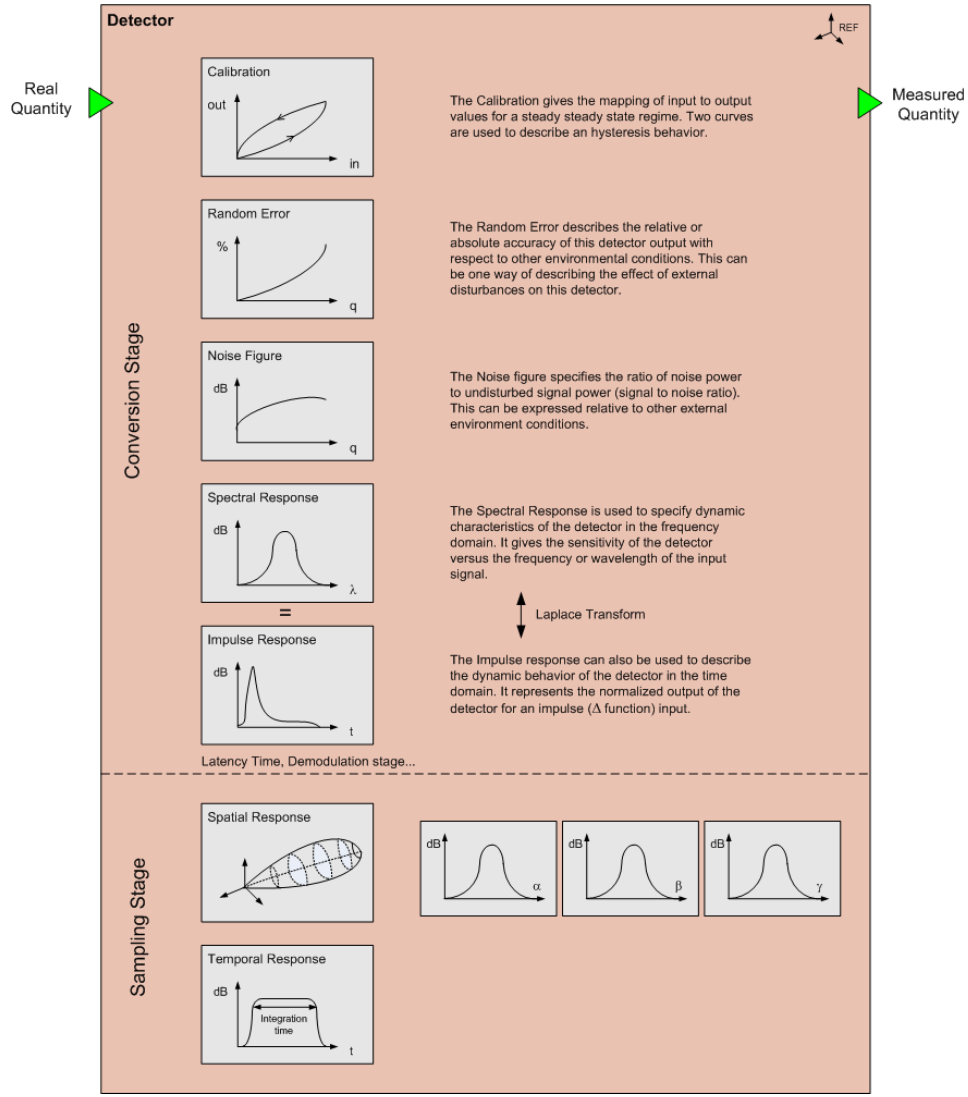
UAH The University of Alabama in Huntsville

- Transducers (detectors, actuators, samplers, etc.)
- Spatial transforms (static and dynamic)
  - Vector, matrix, quaternion operators
  - "Sensor models"
    - scanners, frame cameras, SAR
    - polynomial models (e.g. RPC, RSM)
    - tie point model
  - Orbital models
  - Geospatial transformations (Map projection, datum, coordinate system)
- Digital Signal Processing / image processing modules
- Decimators, interpolators, synchronizers, etc.
- Data readers, writers, and access services
- Human analysts

General Detector Model (Informative)

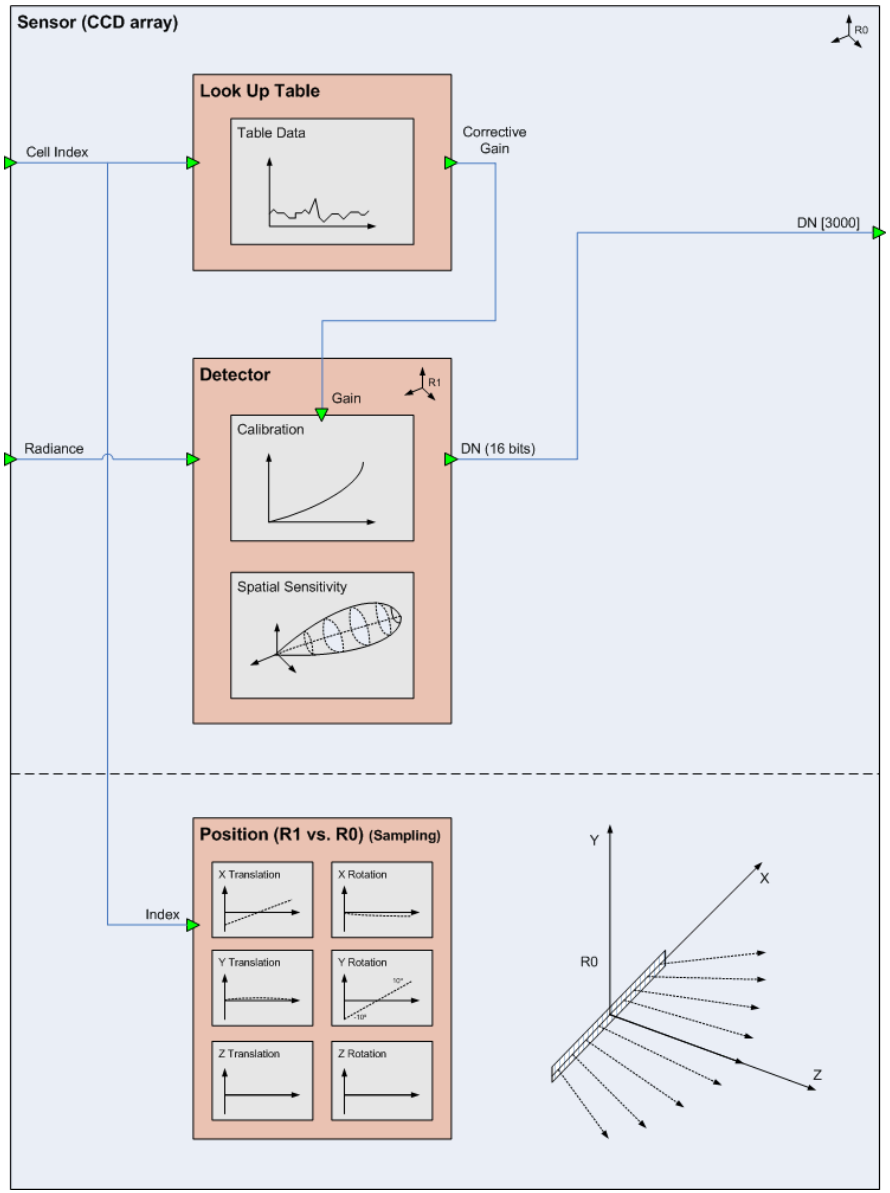
UAH The University of Alabama in Huntsville

- Potentially one of the most complex process models
- All parameters expressed in SensorML as curves
- Describes response of a single detector (e.g. thermometer, pixel, etc.)

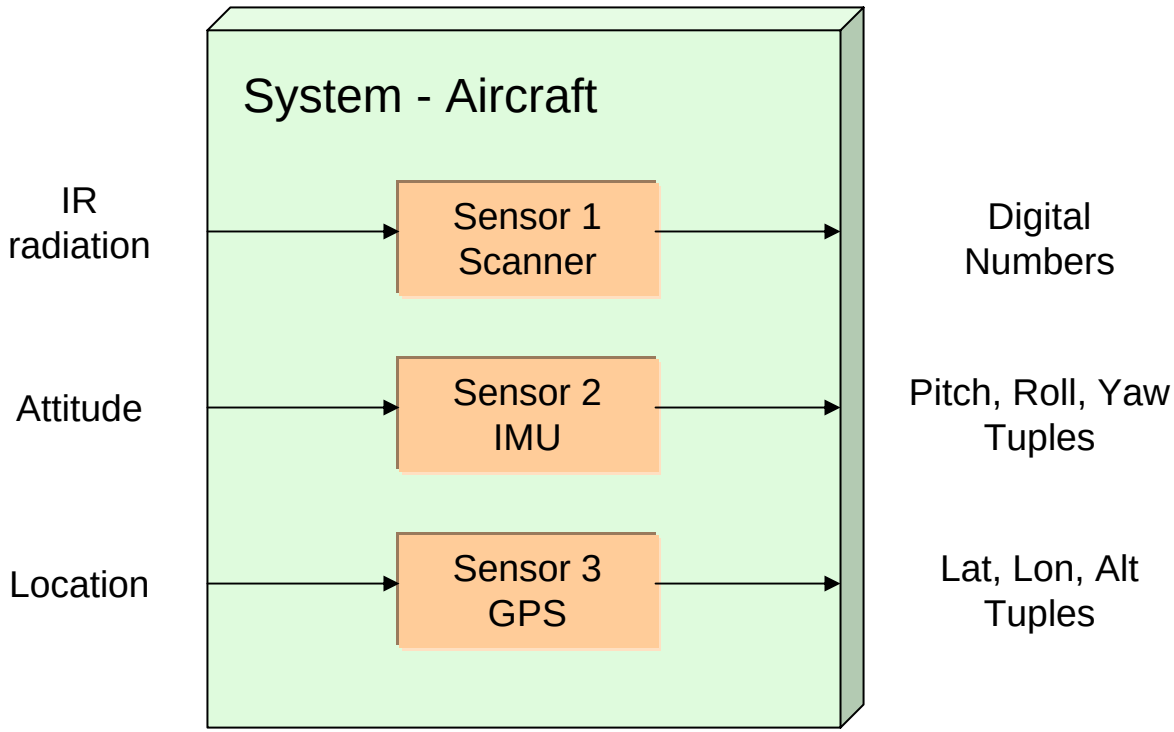


General Detector Array Model (Informative)

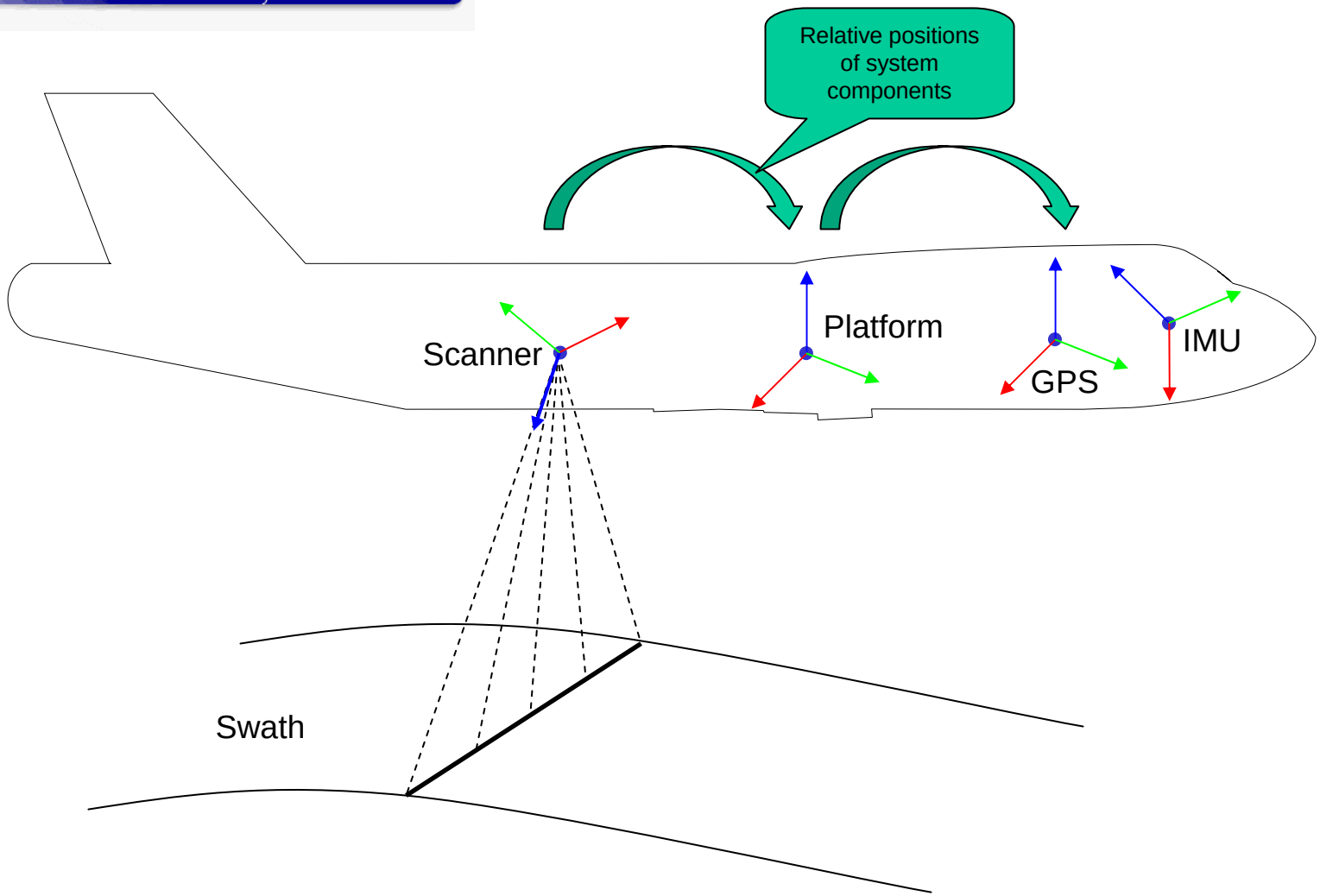
- Detector Array modeled as a System
- Constant properties of individual detectors supported by the Detector ProcessModel
- Variable response of individual detectors supported by index dependent curves (e.g. gain vs. index)
- Spatial variability (e.g. Look ray direction) also supported using index driven curve



SensorML – Sensor Systems



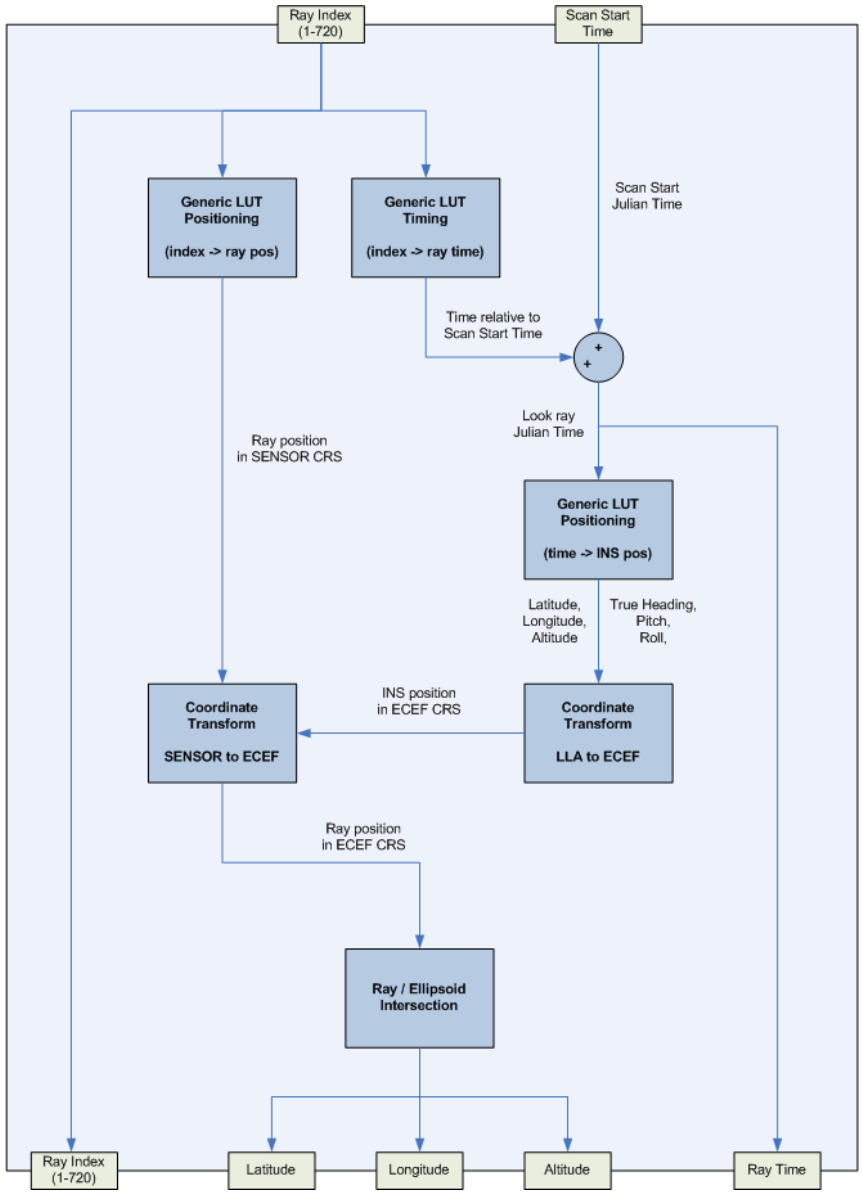
SensorML – Relative Positions



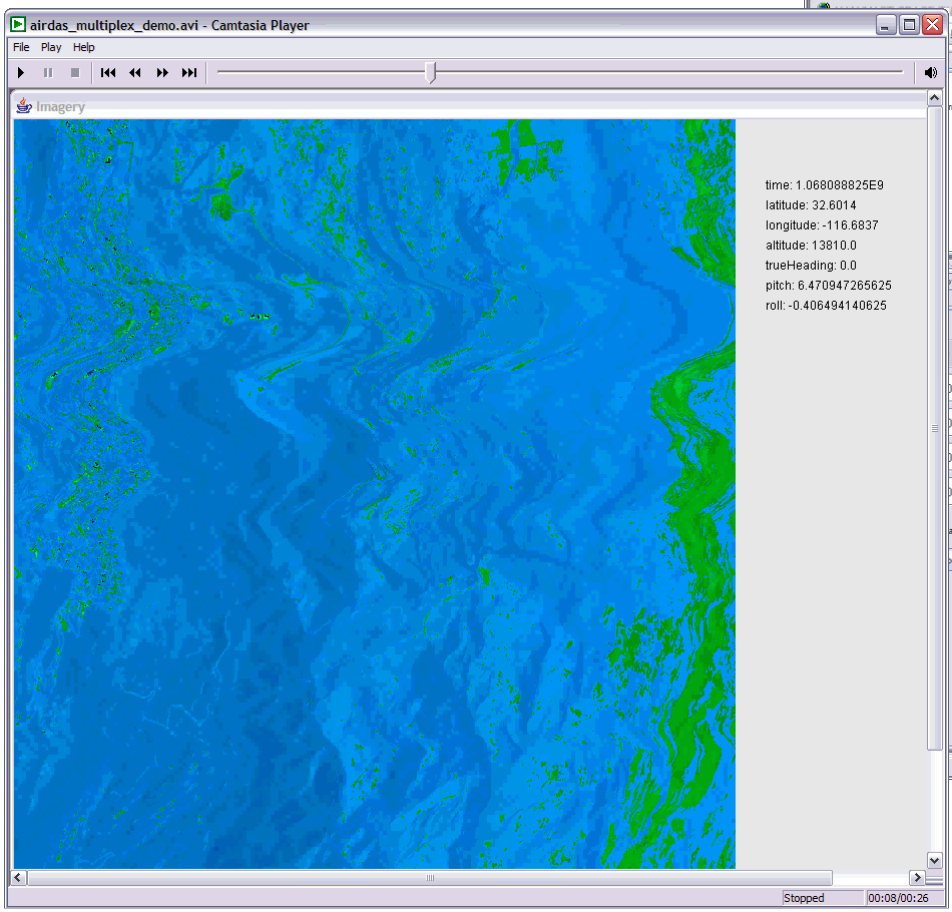
AIRDAS UAV Geolocation Process Chain (Informative)

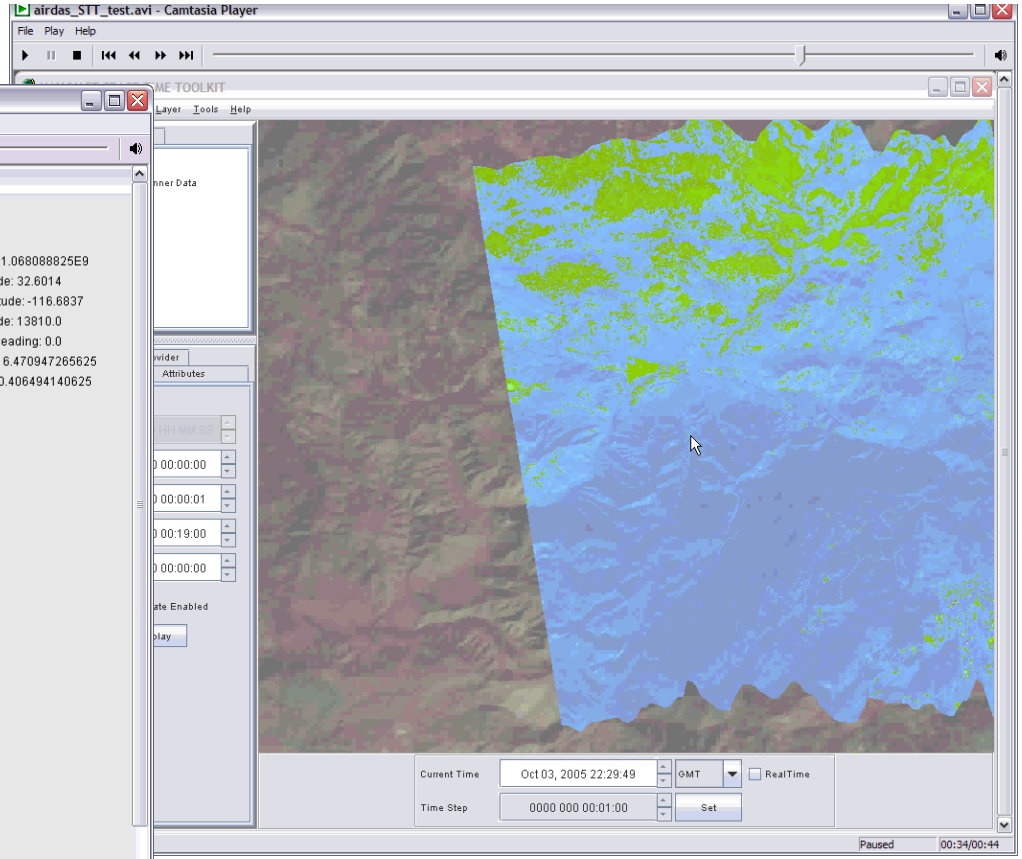
UAH The University of Alabama in Huntsville

- **Inputs:**
  - Ray index
  - Scan start time
- **Outputs:**
  - Ray index
  - Latitude, longitude, altitude
  - Ray time
- **Process Models used**
  - LUTs
    - Index to ray direction
    - Index to ray time offset
    - Time to UAV state
  - Coordinate Transforms
  - Ray-Ellipsoid intersection



AIRDAS UAV Geolocation Process Chain Demo





AIRDAS data stream (consisting of navigation data and 4-band thermal-IR scanline data)

AIRDAS data stream geolocated using SensorML-defined process chain (software has no a priori knowledge of sensor system)

Status of SensorML Specification -1-

- **SensorML Core (Normative)**
  - Core framework models and encoding complete and stable
  - SensorML history
    - Started at UAH in 1998; brought into OGC in 2000
    - Approved as Public Discussion Paper (2002)
    - Approved as Recommended Paper (2004)
    - OGC 05-086 approved as Best Practices Document in Bonn (Nov 2005)
  - Current: document (OGC 05-085r2); schema (2006-02-21)
  - SensorML on track for OGC Technical Specification
    - Gathering and documenting current implementations
    - Approved for RFC in Huntsville TC (March 2006)
    - Approved on July 24 as V0.0 Specification
    - RWG will consider submitted comments and release final V1.0 for approval (60 day vote)

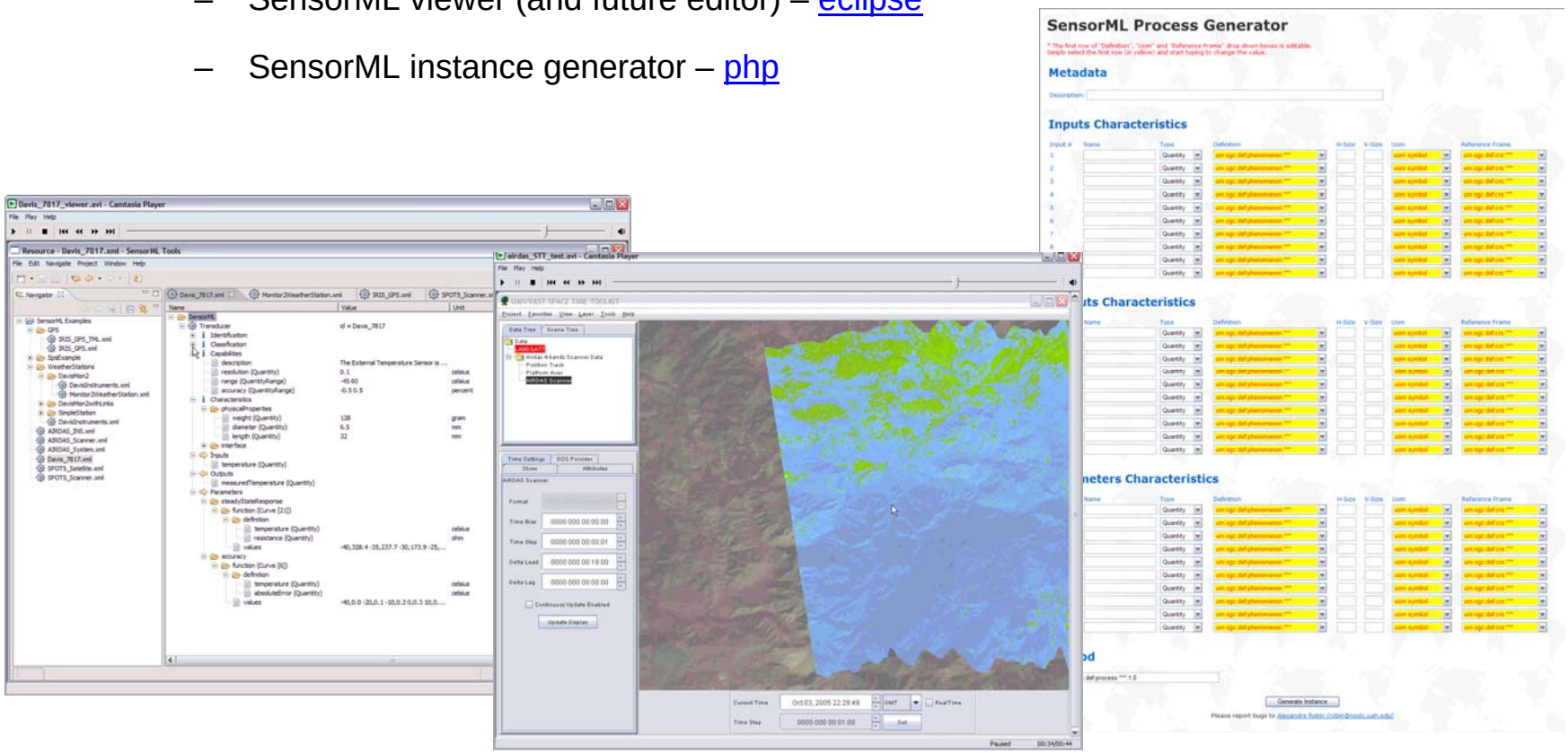
Status of SensorML Specification -2-

UAH The University of Alabama in Huntsville

- **SensorML Forum (Informative)**
  - Established for information exchange (currently 160+ subscribers)
  - http://mail.opengeospatial.org/mailman/listinfo/sensorml
- **SensorML Process Models (Informative)**
  - Have established SourceForge project for ProcessModel submission and download
  - Have developed Open Source SensorML parser and process execution environment
  - Have developed API for Java process model software
- **SensorML White Papers and Tutorials (Informative)**
  - Creating Simple System
  - Creating Process Models
  - SPOT Image System, AIRDAS System Tutorial (coming soon)

Status of SensorML Specification -3-

- **SensorML Open Source Software (Informative)**
  - SensorML parser and chain executor – java
  - SensorML viewer (and future editor) – eclipse
  - SensorML instance generator – php



Recommendations to Net-Ready Sensors Workshop

UAH The University of Alabama in Huntsville

- Consider supporting community models in SensorML
  - Mapping of IEEE P1451 to SensorML is currently underway
  - N42.42 is very compatible with SensorML detector structure
  - Initial investigation indicates CRBN standard is compatible with SensorML
- Advantages
  - SensorML utilizes soft-typing so most mapping primarily involves defining community-approved dictionary (registry) of terms
  - Extensive work and expertise that has been put into community models can be supported in an interdisciplinary (and interoperable) standard
  - Discoverable based on metadata and model characteristics
  - Community Model instances can participate as part of a SensorML processing chain
  - Can include CRBNE/N42.42/IEEE 1451 sensor instances as part of the lineage chain provided by SensorML

UAH The University of Alabama in Huntsville

Sensor Web Enablement





Sensors Are Everywhere








Basic Desires

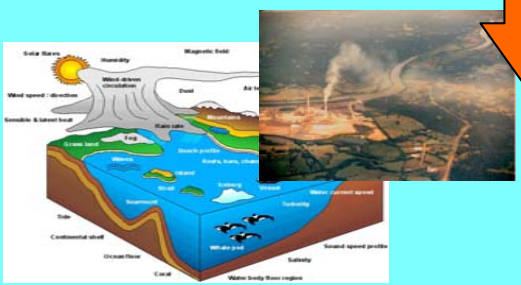
- Quickly discover sensors and sensor data (secure or public) that can meet my needs – location, observables, quality, ability to task
- Obtain sensor information in a standard encoding that is understandable by me and my software
- Readily access sensor observations in a common manner, and in a form specific to my needs
- Task sensors, when possible, to meet my specific needs
- Subscribe to and receive alerts when a sensor measures a particular phenomenon

Sensor Web Enablement Framework

Heterogeneous sensor network

- sparse
- disparate
- mobile/in-situ
- extensible

Models and Simulations



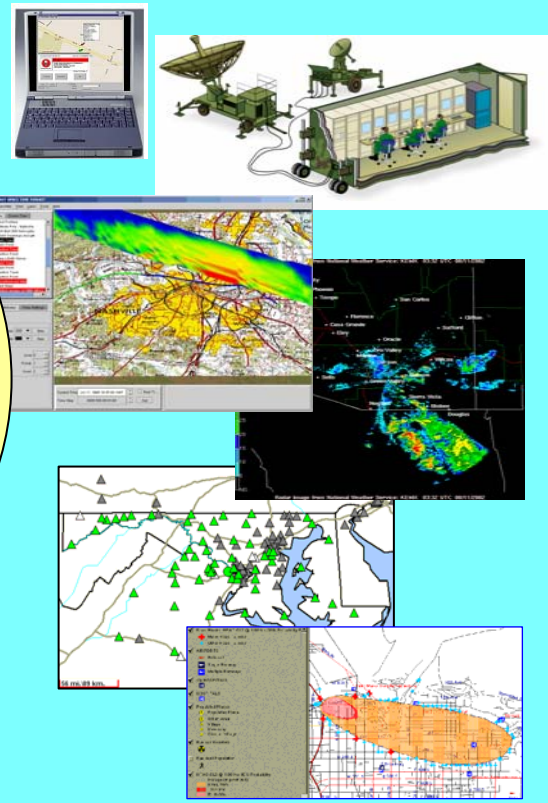
- nested
- national, regional, urban
- adaptable
- data assimilation

Sensor Web Enablement

- discovery
- access
- tasking
- alert notification

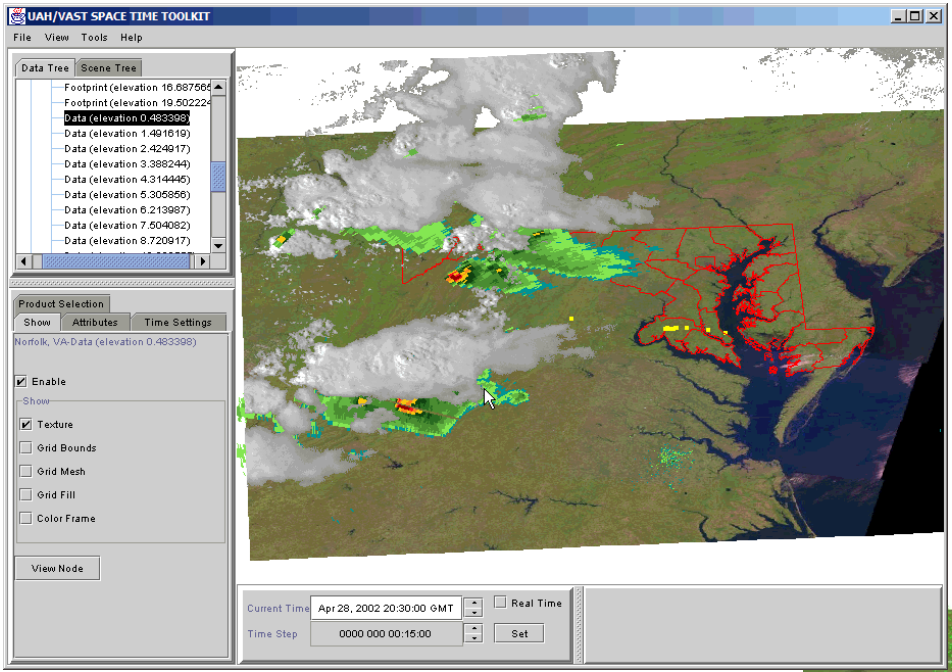
web services and encodings based on Open Standards (OGC, ISO, OASIS, IEEE)

Decision Support Tools

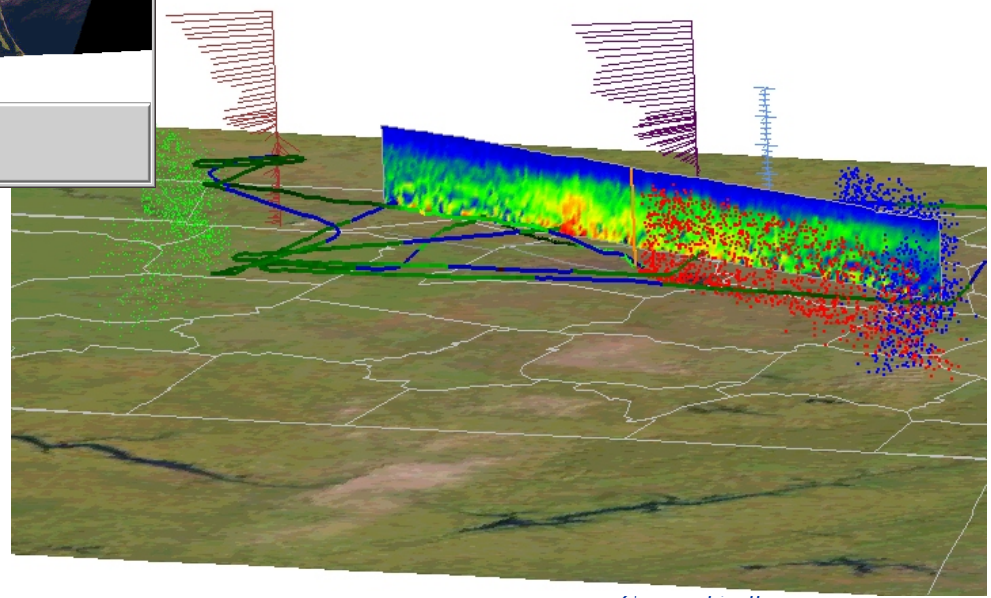


- vendor neutral
- extensive
- flexible
- adaptable

Integration Of Observations From A Variety Of Sensors



We desire the ability to discover and integrate observations from any sensor that meets our needs



OGC

Geographically

Background -1-

1999 - 2000

SensorML initiated at University of Alabama in Huntsville: NASA AIST funding

2001

OGC Web Services Testbed 1.1:

- Sponsors: EPA, NASA, NIMA
- Specs: SOS, O&M, SensorML
- Demo: NYC Terrorist
- Sensors: weather stations, water quality

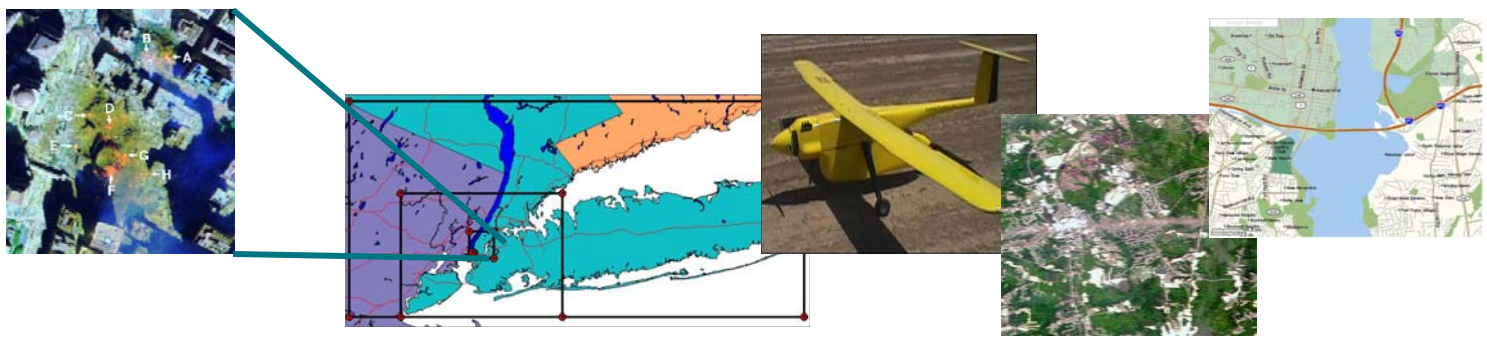
2002

OGC Web Services Testbed 1.2:

- Sponsors: EPA, General Dynamics, NASA, NIMA
- Specs: SOS, O&M, SensorML, SPS, WNS
- Demo: Terrorist, Hazardous Spill and Tornado
- Sensors: weather stations, wind profiler, video, UAV, stream gauges

2003-2004

- Specs advanced through independent R&D efforts in Germany, Australia, Canada and US
- Sensor Web Work Group established
- Specs: SOS, O&M, SensorML, SPS, WNS, SAS
- Sensors: wide variety



Background -2-

OGC Web Services Testbed 3.0:

- Sponsors: NGA, ORNL, LMCO, BAE
- Specs: SOS, O&M, SensorML, SPS, TransducerML
- Demo: Forest Fire in Western US
- Sensors: weather stations, wind profiler, video, UAV, satellite

SAS Interoperabilty Experiment

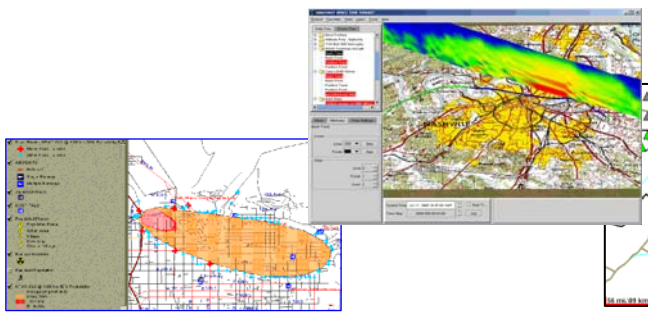
SWE Specifications toward approval:

SensorML – RFC
TransducerML – RFC
SOS – RCF
SPS – RFC
O&M – Best Practices
SAS – Best Practices

OGC Web Services Testbed 4.0:

- Sponsors: NGA, NASA, ORNL, LMCO
- Specs: SOS, O&M, SensorML, SPS, TransducerML, SAS
- Demo: Emergency Hospital
- Sensors: weather stations, wind profiler, video, UAV, satellite

2005 **2006**



SWE Specifications

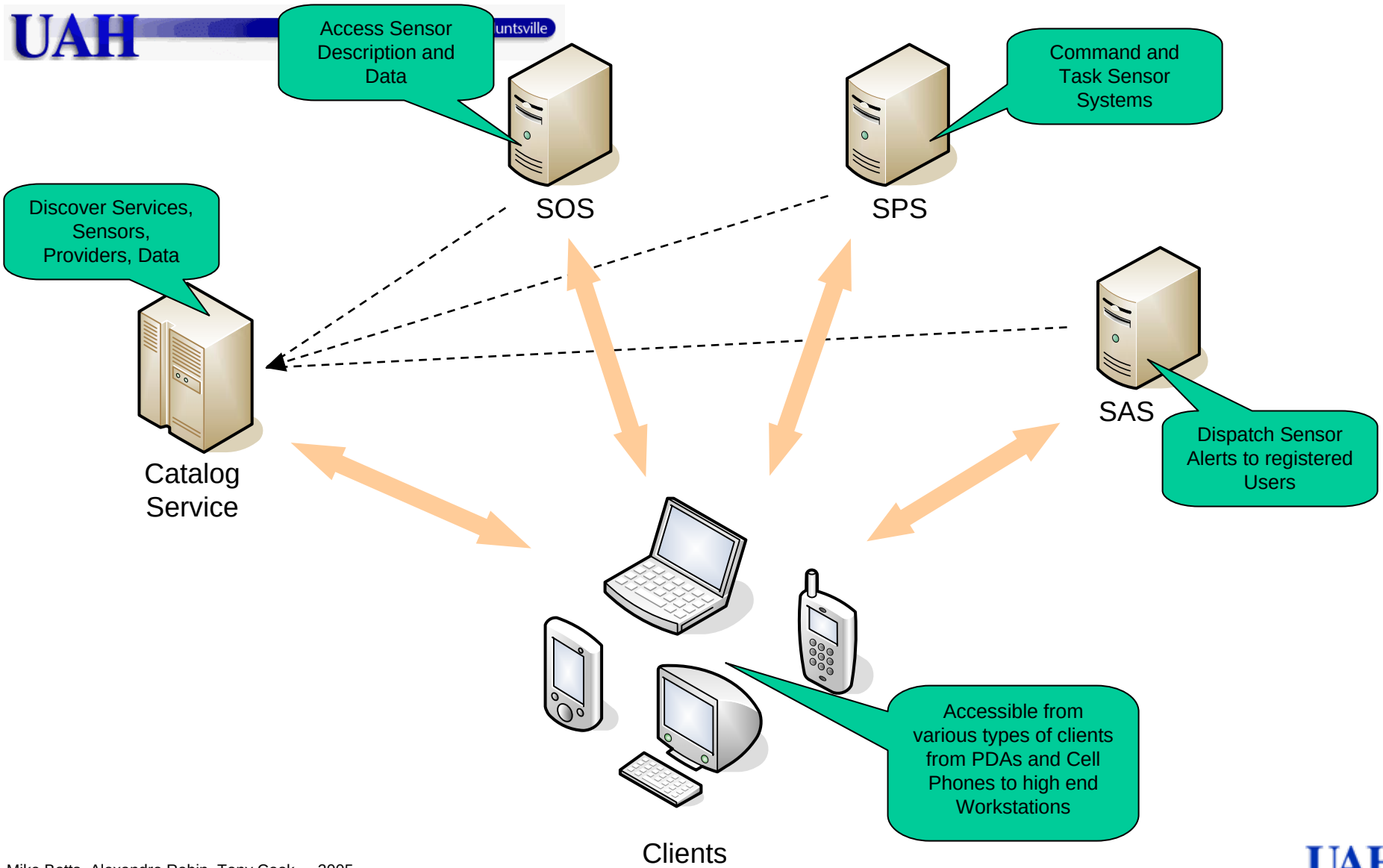
- Information Models and Schema
  - **Sensor Model Language (SensorML) for In-situ and Remote Sensors** - Core models and schema for observation processes: support for sensor components, georegistration, response models, post measurement processing
  - **Observations and Measurements (O&M)** – Core models and schema for observations
  - **TransducerML** – adds system integration and real-time streaming clusters of observations

- Web Services
  - **Sensor Observation Service** - Access Observations for a sensor or sensor constellation, and optionally, the associated sensor and platform data
  - **Sensor Alert Service** – Subscribe to alerts based upon sensor observations
  - **Sensor Planning Service** – Request collection feasibility and task sensor system for desired observations
  - **Web Notification Service** –Manage message dialogue between client and Web service(s) for long duration (asynchronous) processes
  - **Sensor Registries** – Discover sensors and sensor observations

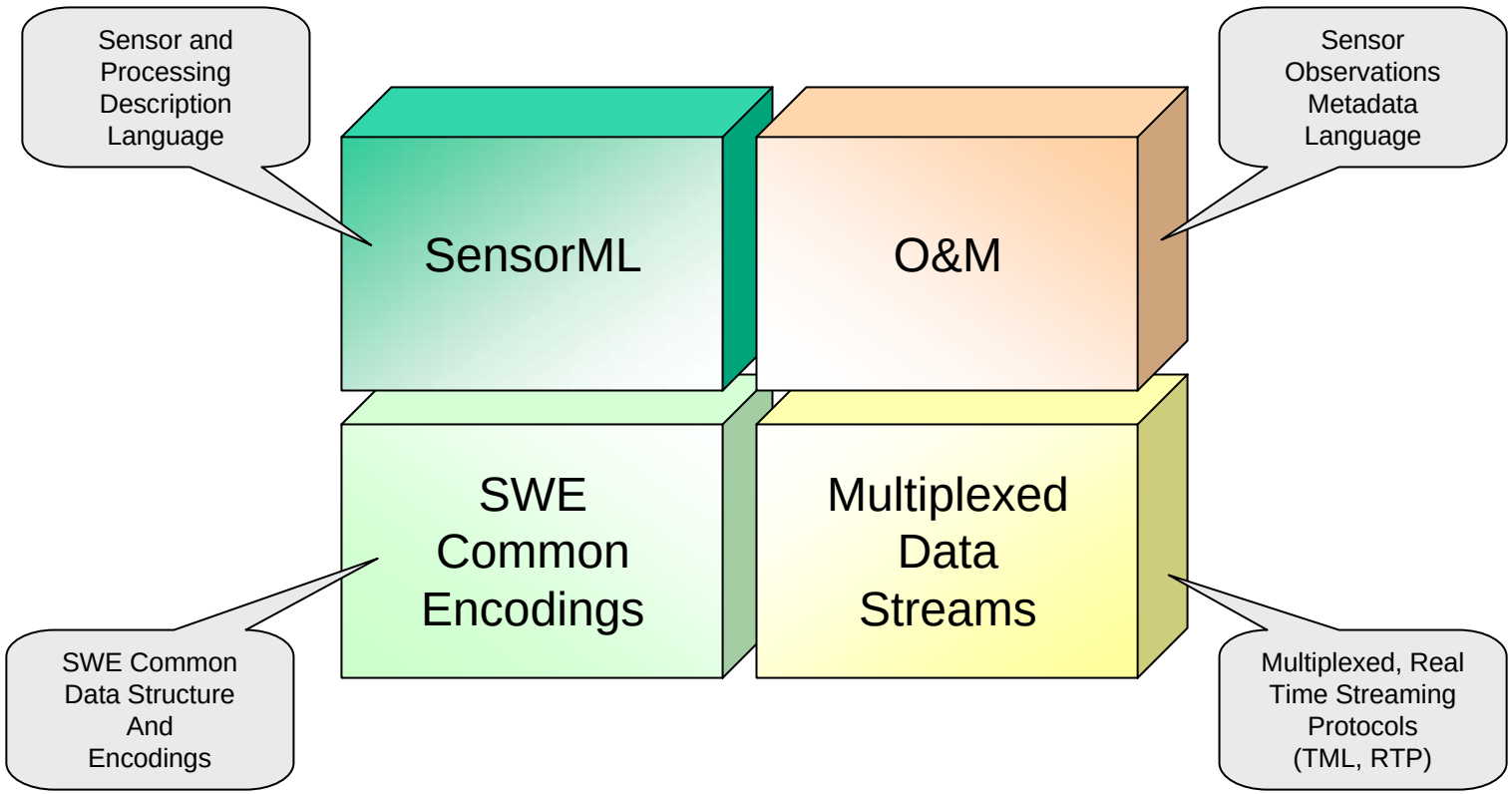
OGC Sensor Web Enablement -4-

- Sensor Web Enablement – Potential Harmonizations
  - **OASIS Common Alert Protocol (CAP)** – being considered as an encoding or portrayal of sensor alerts in SAS
  - **OASIS EDXL** – XML “envelope” for alerts
  - **IEEE P1451** – provides “plug-n-play” capabilities for sensors; working toward automatic conversion between 1451 and SensorML/ TML/ O&M
  - **CBRNE** and **N42.42** – Working on examples supporting these in SensorML

SWE Components – Web Services



SWE Components - Languages



Example SOS Requests and Common Observations

UAH The University of Alabama in Huntsville

Weather SOS Requests

GET Request for Weather SOS Capabilities

http://vast.uah.edu:8080/ows/weather?request=GetCapabilities

GET Request for Weather Data measurements

http://vast.uah.edu:8080/ows/weather?request=GetObservation&offering=WEATHER_DATA&time=2004-04-01T05:00:00Z/2004-04-01T06:00:00Z&format=application/com-xml

Airdas SOS Requests

POST Requests Examples - Here

GET Request for Airdas SOS Capabilities http://vast.uah.edu:8080/ows/airdas?request=GetCapabilities

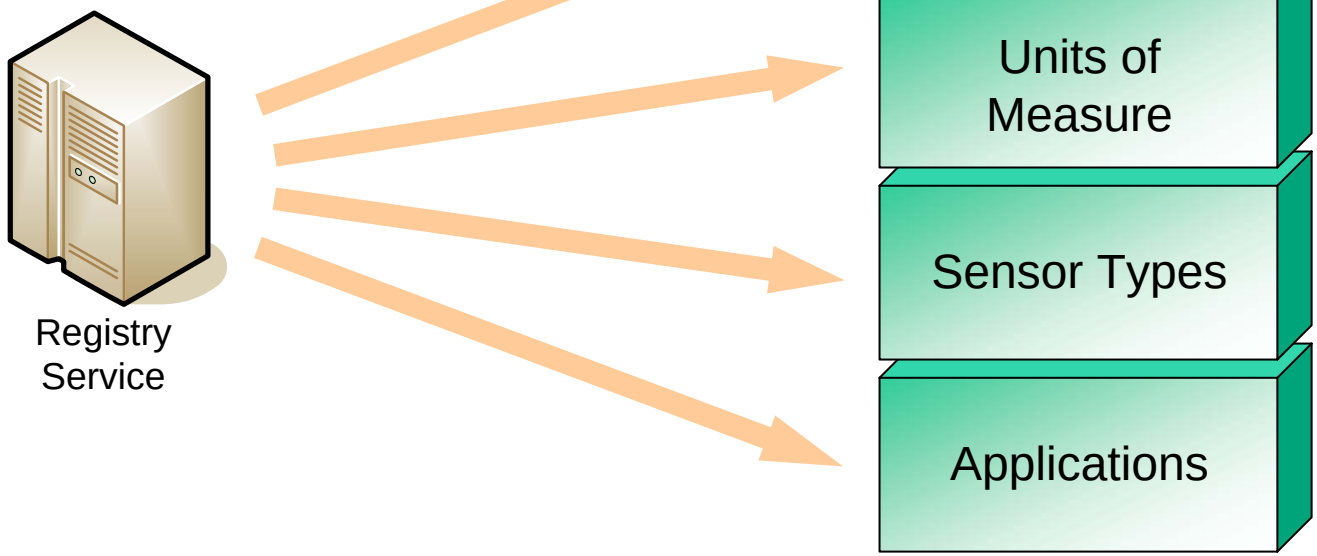
GET Request for Airdas Navigation Data measurements

http://vast.uah.edu:8080/ows/airdas?request=GetObservation&offering=AIRDAS_NAV&time=2003-11-05T19:30:00-08:00/2003-11-05T19:40:00-08:00&format=application/com-xml

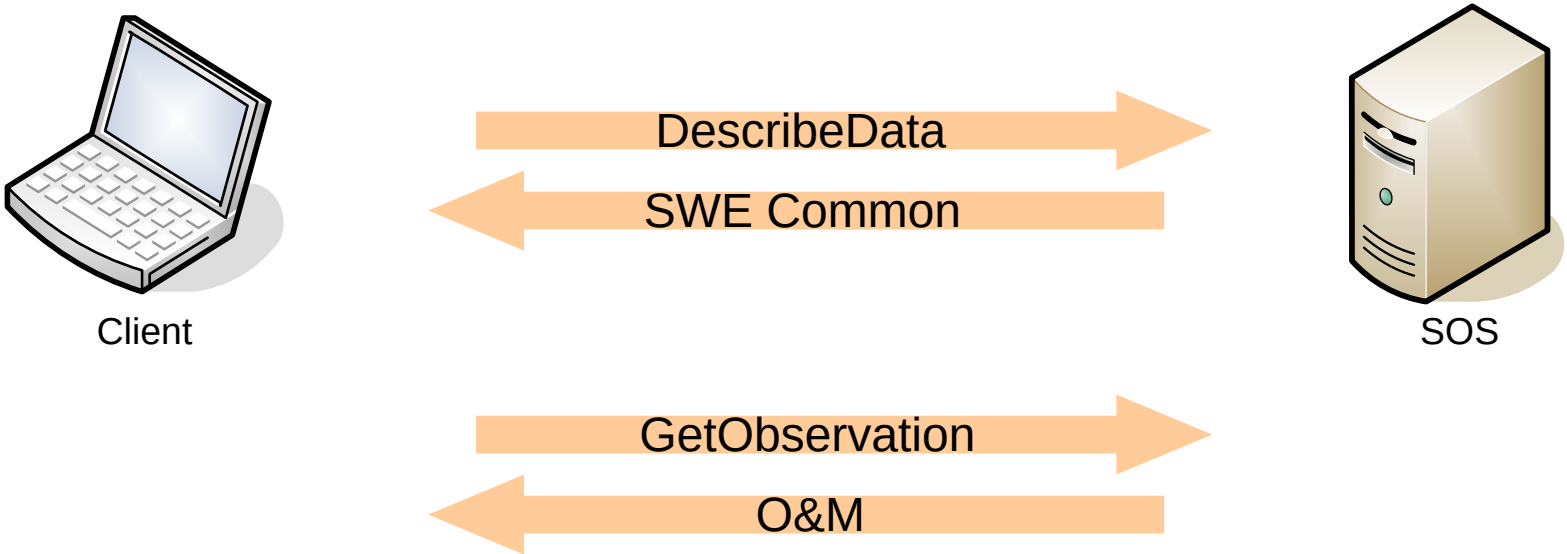
GET Request for Airdas Scanner Data measurements

http://vast.uah.edu:8080/ows/airdas?request=GetObservation&offering=AIRDAS_SCAN&time=2003-11-05T19:19:00-08:00/2003-11-05T19:19:10-08:00&format=application/com-binary-base64

SWE Components - Dictionaries



SWE – Sensor Observation Service



SWE – Sensor Planning Service

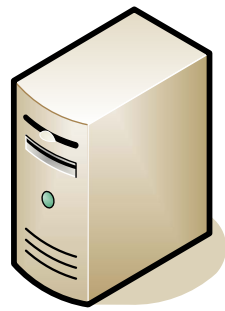
Client

DescribeSensor

SensorML

GetTaskableProperties

SWE Common

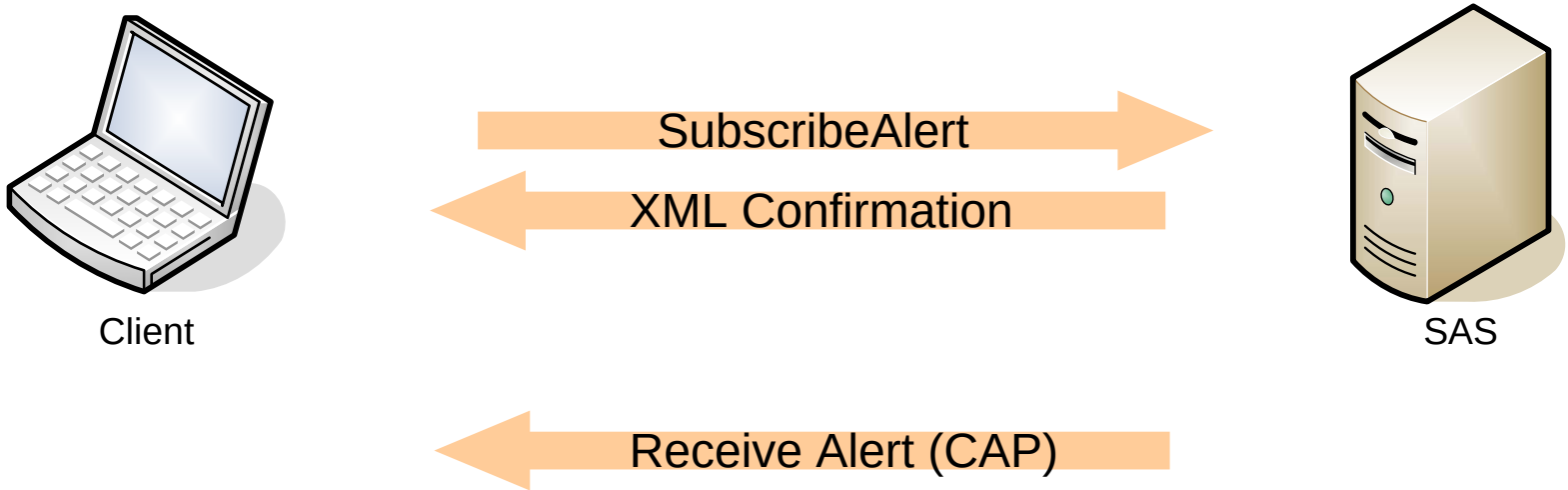


SPS

SendTask

XML Confirmation

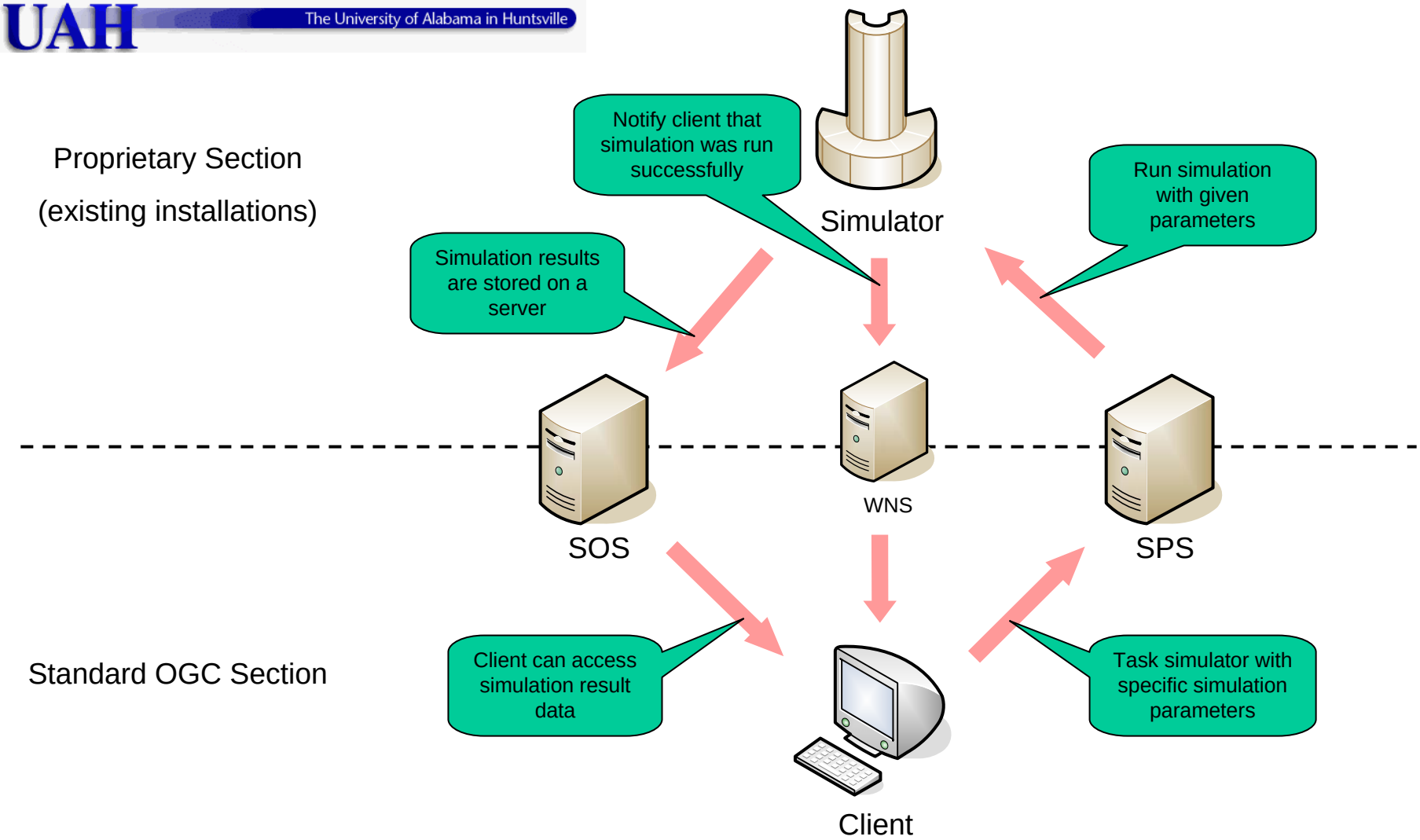
SWE – Sensor Alert Service

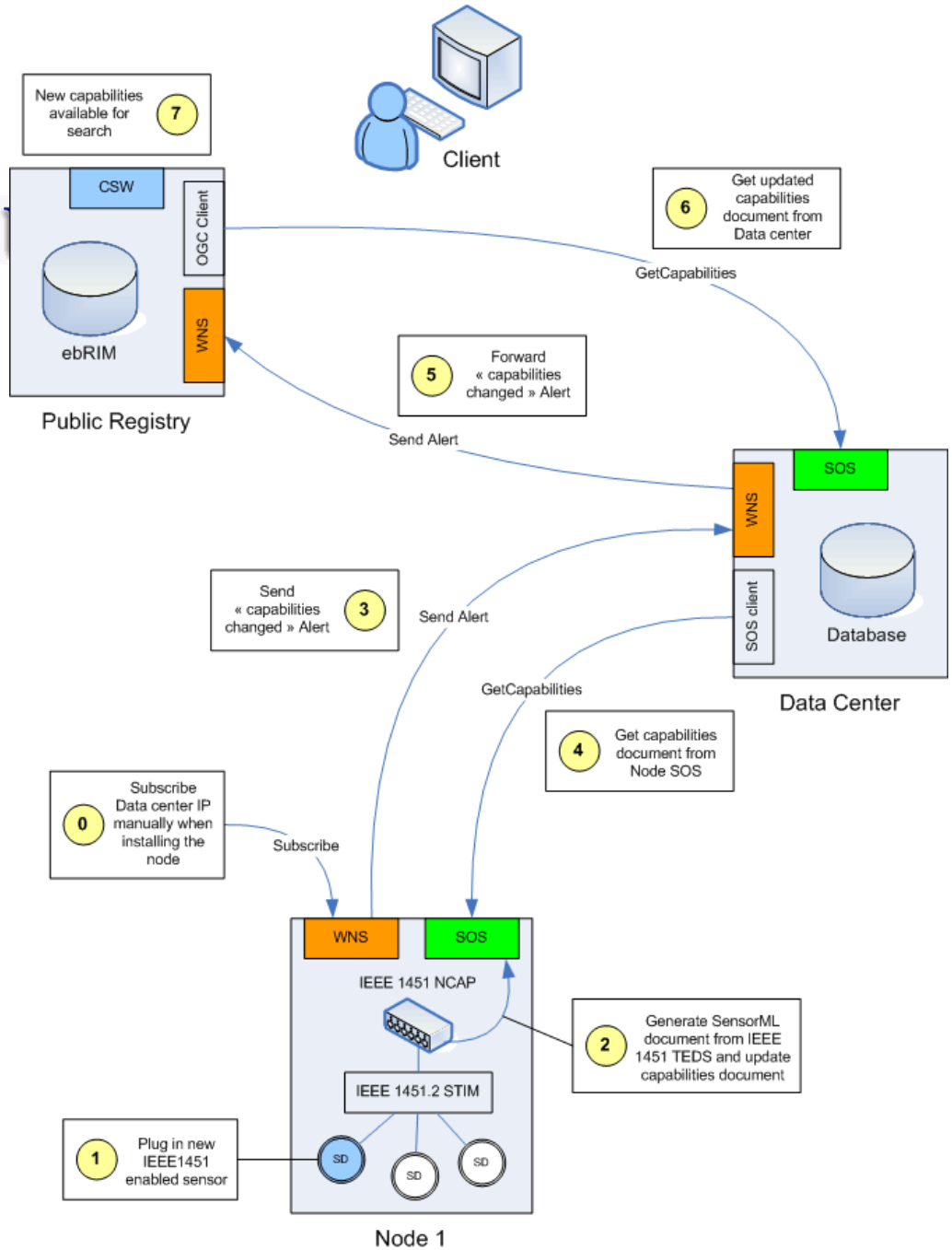


Status

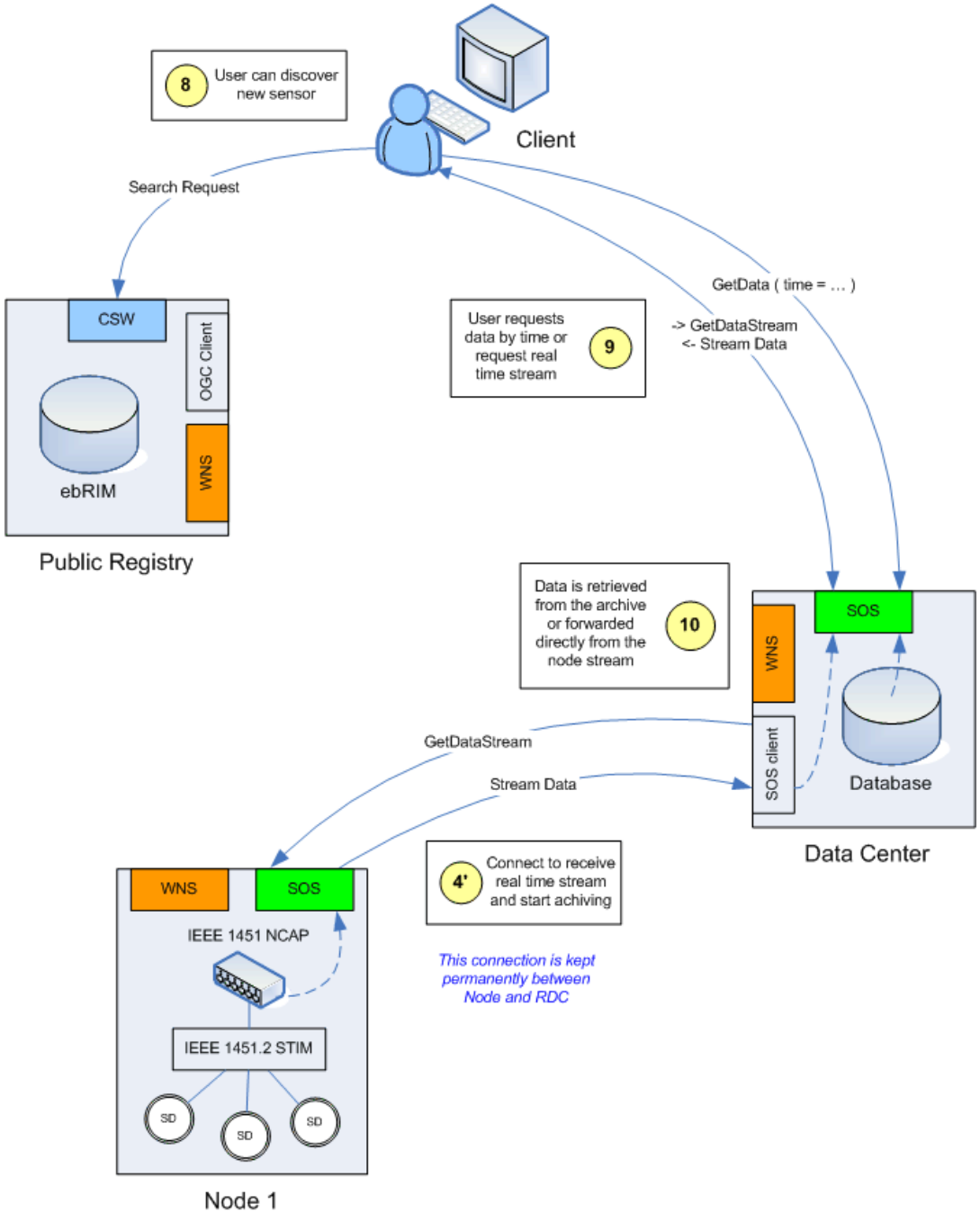
- Current specs are in various stages
  - SensorML – Past RFC; Approved as Version 0.0
  - TransducerML – RFC
  - O&M – Best Practices Paper (soon RFC)
  - WNS – Public Discussion Paper
  - SOS – RFC
  - SPS – RFC
  - SAS – Best Practices Paper (soon RFC)
- OGC Web Services (OWS) 4.0 includes thread for Sensor Web Enablement
  - Execution: June 2006 to December 2006

Integration of Simulation Capabilities

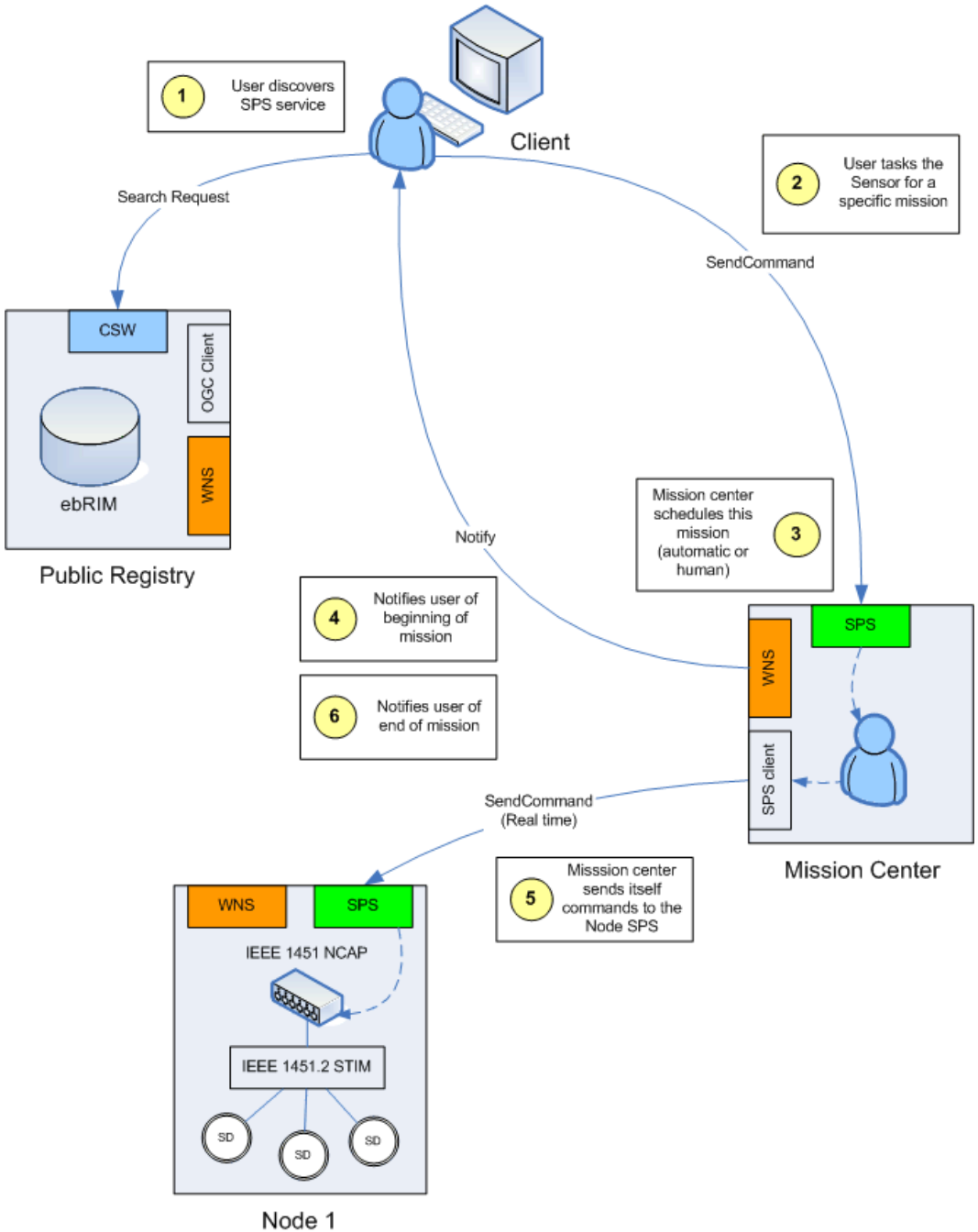




ORNL
Plug-n-Play



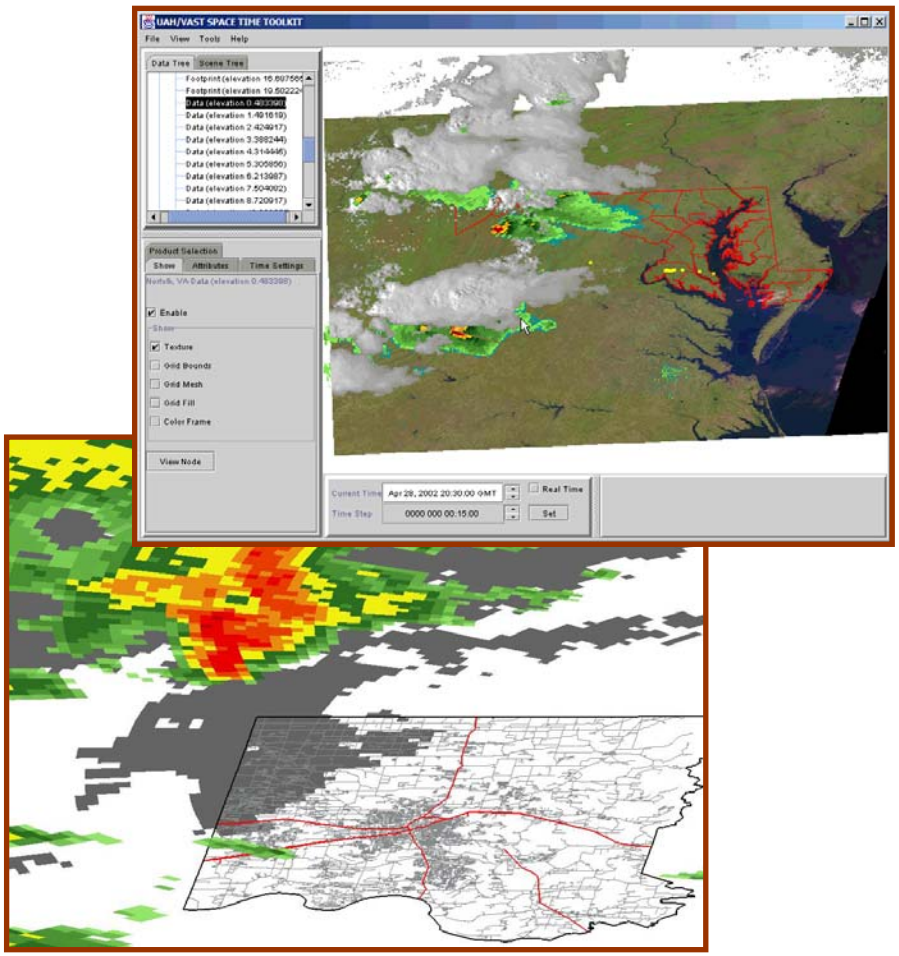
ORNL Client Access

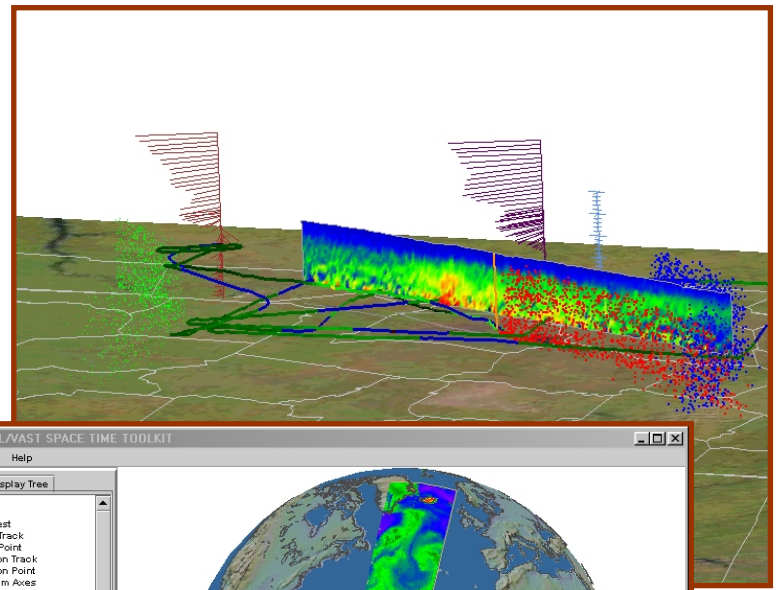


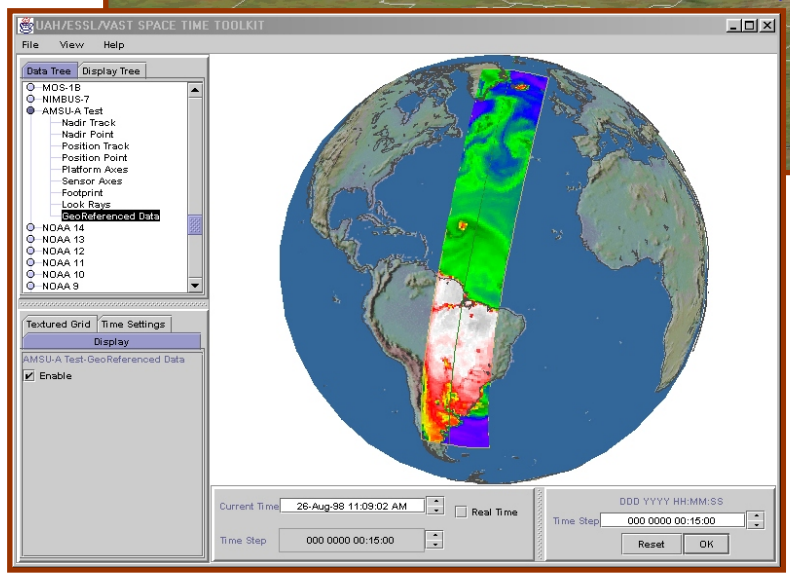
ORNL
Client Tasking

Space Time Toolkit
Sample Applications

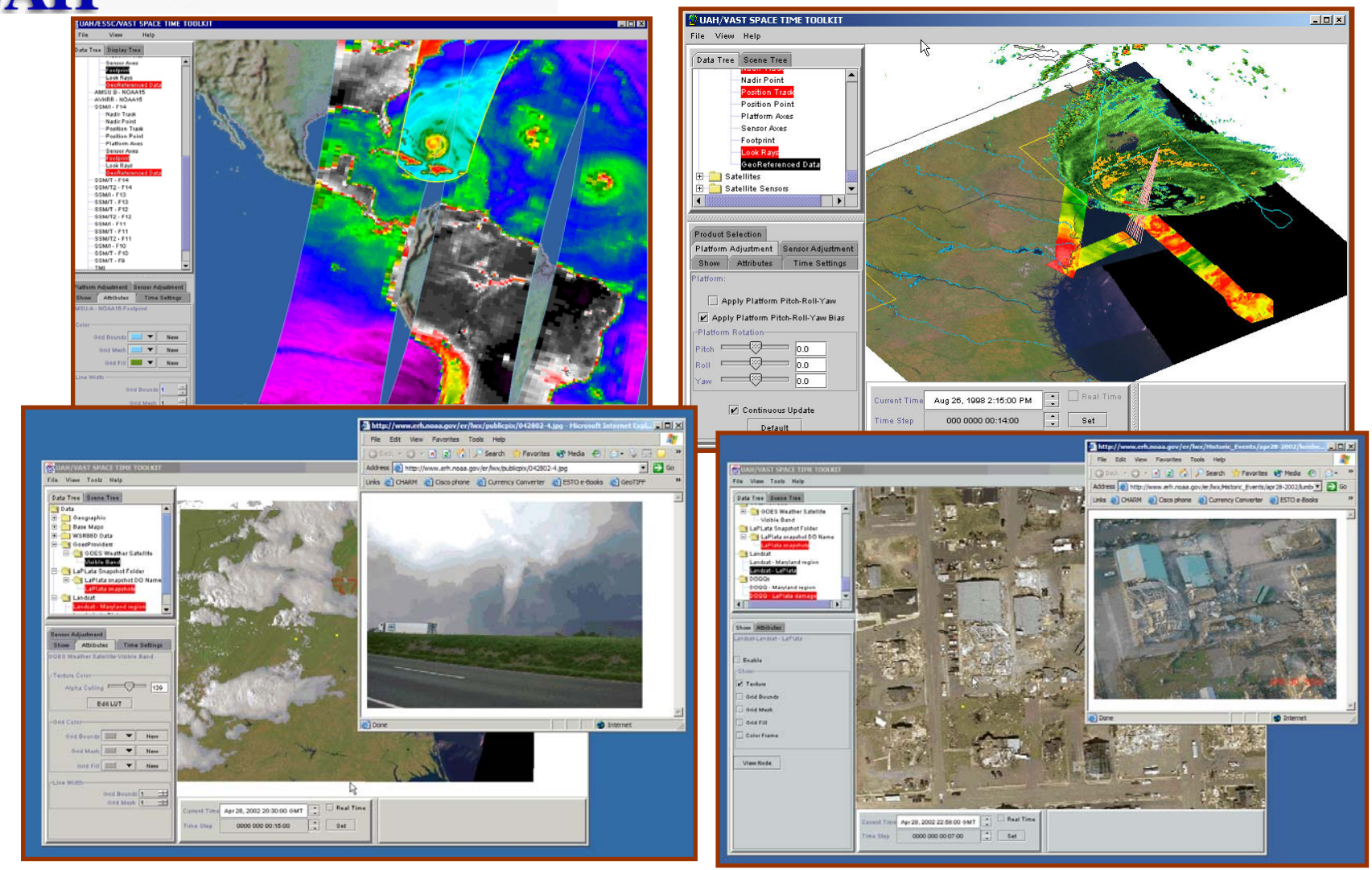
UAH The University of Alabama in Huntsville







Space Time Toolkit
Sample Applications -2-



Relevant Links

- **OpenGeospatial Consortium –**

  http://www.opengeospatial.org

- **Sensor Web Enablement –**

  http://www.opengeospatial.org/projects/groups/sensorweb

- **SensorML –**

  http://vast.uah.edu/SensorML

- **Space Time Toolkit –**

  http://vast.uah.edu/SpaceTimeToolkit